STATISTICAL SIGNIFICANCE FOR DNA MOTIF DISCOVERY

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Patrick P. Ng

May 2011

STATISTICAL SIGNIFICANCE FOR DNA MOTIF DISCOVERY

Patrick P. Ng, Ph.D.

Cornell University 2011

The identification of transcription factor binding sites, and of *cis*-regulatory elements in general, is an important step in understanding the regulation of gene expression. To address this need, many motif-finding tools have been described that can find short sequence motifs given only an input set of sequences. In this dissertation, we will begin by discussing why a reliable significance evaluation should be considered an essential component of any motif finder. We will introduce a biologically realistic method to estimate the reported motif's statistical significance based on a novel 3-Gamma approximation scheme. Furthermore, we show how the reliability of the significance evaluation can be further improved by incorporating local base composition information to its null model. We then demonstrate its reliability by applying GIMSAN/MOTISAN — *de novo* motif finding tool that incorporates this novel significance evaluation technique — to a well-studied set of *Saccharomyces cerevisiae* motif input data. Our results also reveal that an ensemble method based on our significance evaluation can substantially improve the actual motif finding task.

Finally we will present ALICO (Alignment Constrained) null set generator: a framework to generate randomized versions of an input multiple sequence alignment that preserve some of its crucial features including its dependence structure. In particular, we will show that, on average, ALICO samples approximately preserve the PIDs (percent identities) between every pair of input sequences as well as the average Markov model composition. We will demonstrate its utility in phylo-

genetic motif finders — motif finding tools that leverage conservation information — in terms of both reliability of statistical significance and improvement of motif finding task through ensemble method.

BIOGRAPHICAL SKETCH

Patrick Ng spent his early childhood in Hong Kong. At the age of nine, he moved to the Bensonhurst neighborhood of Brooklyn, New York. Patrick attended Brooklyn Technical High School for four years. He then received a B.S. and M.Eng. in Computer Science from Cornell University in May 2005. Patrick will be receiving a Ph.D. in Computer Science from Cornell University in May 2011.

ACKNOWLEDGEMENTS

First and foremost, I thank my advisor Uri Keich for his guidance and wisdom. It has been a privilege working with him. I am especially appreciative of his patience in the past 2 years. I am also indebted to him for introducing me to the interesting and exciting area of computational biology.

I thank Eric Friedman for his encouragement and advice, and I am grateful to have the opportunity to work with him when I was an undergraduate. I am fortunate to have James Booth in my committee, and I thank him for his advice and insightful discussions about my thesis research.

I would like to thank Niranjan Nagarajan for our collaboration together during my early years in graduate school. Also, I acknowledge the Tri-I CBM program for its financial support. More importantly, many thanks to the wonderful community of fellow students and faculties in the CBM program for being supportive and providing a great environment to communicate and do research.

I express my gratitude to my officemates who have kept me sane for the past five years, especially Peter Majek, Kelvin So, and Mohamed. I am sincerely thankful for having friends who have given me so much encouragement and support over the years — some of whom I had the joyous moments of traveling together, poker, board games, midnight shopping at Wegmans, snowboarding, traying down Libe slope, and allowing me to stay at their homes when I needed to get away from Ithaca. In particular, I would like to thank: Kang, Shuwah, Collin, Siukei, Kingyin, Hester, Ben Chan, Kelvin Leung, Betty, Melanie, Julia, Euan, Jia Chen, Kelly, Yancy, Chun-Nam, Daniel, Vincent Ng, and Arielle.

Finally, I would like to thank my parents and my brother Tony for their constant support and love. I would like to thank my mom for her daily phone call of encouragement and her exemplification of hard work and perseverance. I thank

my dad for his diligence in everything he does and for teaching me elementary algebra, which is something I still use to this very day.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Ever since the invention of computers, the automation of pattern searching within large-scale data has been ubiquitous in the field of computer science. Computer scientists invented data structures such as suffix trees to perform string searches in a plain text file, and then moved on to more sophisticated types of pattern searching on different domains, such as databases, file systems, world wide web, and social networks. Almost a decade ago, with the sequencing of the human genome, we became the first species to be able to peer into the "blueprint" of our own genetic makeup. Thus it has offered scientists to perform automated pattern searching for biologically-relevant elements across the 3 billion base pairs of the human genome. This thesis will introduce new algorithms and statistical methods for one such class of searches within the genome called *motif discovery*.

1.1 What is a motif?

The word *motif* is often used to describe a short recurring central theme or pattern in art and music. In biology, a DNA motif is an over-represented recurring nucleotide pattern that has biological significance. A motif is typically a short pattern, consisting of 5 to 20 nucleotides long, within a set of much longer background sequences. It is an element that recurs within a set of sequences that share a common biological function. Thus motifs are often used to model and identify transcription factor binding sites and *cis*-regulatory elements (Hertz et al., 1990; Lawrence et al., 1993; Stormo, 2000; Harbison et al., 2004).

What exactly is a nucleotide pattern? The most intuitive and simple pattern would be a nucleotide string or word. For example, the TATA-binding protein binds to a DNA sequence of `TATA`, which are found in the promoter regions of most eukaryotic genes. The surrounding or flanking nucleotides of the `TATA` string within the promoter regions is often different, thus `TATA` is an over-represented string. However, there exists motifs that are less conserved than this. Specifically, there are biologically-relevant DNA patterns that are not exact recurring strings, but rather there are some variations between the recurring DNA sequences within the pattern. As we will see later in this chapter, there are different representations for these degenerate motifs. We will also later see how to evaluate and model the *degeneracy* of these motifs, so that we can differentiate between real motifs and background sequences.

1.2 Motif-finding and motif representation

There are several ways to represent a motif. One method is obviously the raw sequences of each site of a motif. This is, however, often difficult to interpret and understand because a motif of fixed width w with n instances would require an $n \times w$ matrix to represent it. In this section, we will describe two methods — consensus and PWM (probability weight matrix) — to represent a motif that is more concise than the raw sequences of each site. But firstly, we will go over some formal definitions and notations that will be useful to us.

Given a set of N sequences $S = \{S^1, S^2, \ldots, S^N\}$, the formal definition of the *(de novo) Motif Finding Problem* is to find the set of starting positions within each sequence that corresponds to the location of an *unknown* motif of width w (Hertz et al., 1990; Lawrence et al., 1993; Bailey and Elkan, 1994; Keich and Pevzner, 2002a). If Σ is the set of alphabet, then the sequence $S^i \in \Sigma^{l_i}$ where l_i is the length of each sequence S^i. Since we are only focusing on DNA motifs in this thesis, our alphabet set is $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$. Let Y_i be the starting position of the motif in sequence S_i. Thus the set of motif instances are $\{S^i_{Y_i} S^i_{Y_i+1} \cdots S^i_{Y_i+w-1}\}_{1 \le i \le N}$. From here on, we will use the shorthand notation $\{S^i_{[Y_i:Y_i+w-1]}\}_{1 \le i \le N}$ for the set of motif instances.

Consensus motif uses a single string of width w to represent a motif of width w. The consensus motif can be constructed by taking the most occurring nucleotide/character of each motif position. Specifically, the j-th position of the consensus motif can be given by:

$$\text{argmax}_{b \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} \sum_{1 \le i \le N} 1_{S^i_{Y_i+j-1}=b}$$

$$\begin{array}{c} A \\ C \\ G \\ T \end{array} \begin{pmatrix} 0.2 & 0.2 & 0.6 & 0.0 & 0.2 \\ 0.6 & 0.8 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.2 & 0.0 & 0.4 & 0.0 & 0.8 \end{pmatrix}$$

Figure 1.1: Motif PWM estimated from motif instances {`CCAGA, CAAGT, CCTGT, ACTGT, TCAGT`}

For example, given motif instances $\{\texttt{CCAGA}, \texttt{CAAGT}, \texttt{CCTGT}, \texttt{ACTGT}, \texttt{TCAGT}\}$, the consensus motif would be `CCAGT`. Note that the consensus motif in this example has at most two mutations from any of the motif instances. Hence under the consensus motif representation, this example has consensus `CCAGT` with at most two mutations. In addition, other IUPAC or "wildcard" symbols are often used in consensus motif. For example, the character `R` represents purine, which can either be the nucleotide `A` or `G`.

Another method to represent a motif is using a *probability weight matrix* (PWM), also known as *profile* (see Figure 1.1). A PWM Θ is a $4 \times w$ matrix where Θ_{bj} is the frequency of nucleotide b at the j-th position of the motif. It can be estimated from motif locations Y by

$$\widehat{\Theta_{bj}} = \frac{1}{N} \sum_{1 \le i \le N} \mathbf{1}(S^i_{Y_i+j-1} = b) \tag{1.1}$$

where $\mathbf{1}(\cdot)$ is the indicator function.

We will often refer the j-th position of a motif $\Theta_{\cdot j}$ as a *motif column*. One major disadvantage of this model is that it does not capture the dependency between motif columns. In particular, the motif columns are assumed to be independent

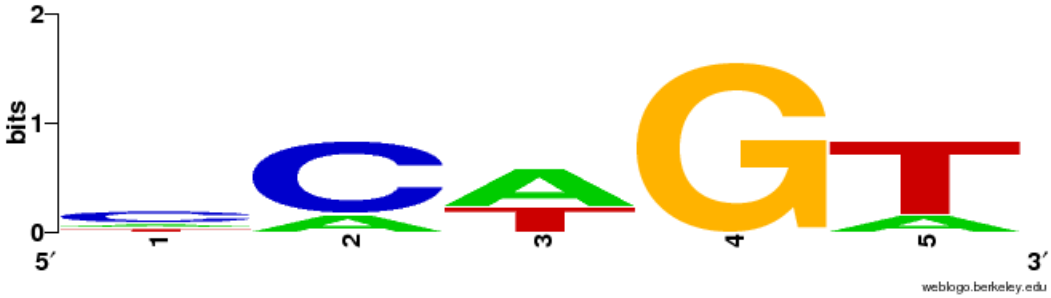


Figure 1.2: WebLogo estimated from motif instances {`CCAGA, CAAGT, CCTGT, ACTGT, TCAGT`}

from each other[a]. In Section 7.3, we will discuss a method to detect dependencies between a pair of motif columns given a set of motif instances.

A popular tool to visualize a motif is using a Weblogo (Crooks et al., 2004) (see Figure 1.2). The height of each bar is the information content of a motif column, measured in bits. The information content of the j-th column of a motif is given by:

$$I_\Theta(j) := 2 + \sum_{b \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} \Theta_{bj} \log_2 \Theta_{bj} \tag{1.2}$$

If a motif column has uniformly distributed frequency across all nucleotides (i.e. $[0.25, 0.25, 0.25, 0.25]^T$), then its information content is 0 bits. An information content of 2 bits is given to a motif column where a specific nucleotide is completely conserved, such as the 4^{th} position in Figure 1.2. The height of each alphabet within the bar is simply its fractional information content. As we will see in the Chapter 2, the information content is the basis for scoring a motif.

[a]Unfortunately modeling motifs using Markov models is not practical for *de novo* motif searches because the sheer number of parameters needed to estimate for such a model.

1.3 Explosion on the number of motif finders

There has been an explosion of the number of available motif finders in the last decade. In (Sandve and Drablos, 2006), the authors tallied over 119 finders. These motif-finders used different approaches to search and evaluate motifs. Currently the differences between motif finders often fall into several categories:

- Motif representation (e.g. PWM, consensus sequence, regular expressions)
- Motif scoring (e.g. likelihood ratio, enrichment score)
- Optimization method (e.g. Gibbs sampling, Expectation-Maximization, branch-and-bound algorithm)
- Motif types (e.g. dimer, microRNA motifs, palindrome, gapped motifs)
- Auxiliary input data (e.g. conservation, nucleosome positioning, ChIP-chip binding signals, ChIP-seq peaks)

As we have discussed in Section 1.2, there are different representations for motifs and thus finders vary in their method to represent them within the algorithm. For example, PatternBranching (Price et al., 2003) used consensus sequences to model motifs, while BioProspector (Liu et al., 2001), MEME (Bailey and Elkan, 1995) and AlignACE (Neuwald et al., 1995) used a PWM model. As for motif scoring, we will discuss various scoring metrics in details in Chapter 2. In Section 3, we will discuss two classes of algorithms that can be used to optimize motif scoring metrics: Gibbs sampling and EM algorithm.

Finally, some motif finders are designed to use other auxiliary input information besides the set of nucleotide sequences to enhance their ability to perform the motif finding task. For example since a higher quantitative binding signal in

ChIP-chip is correlated with the likelihood of a motif site, (Bussemaker et al., 2001; Eden et al., 2007) used the binding signal to improve the sensitivity of their motif finding algorithm. Likewise, with the recent advances in cheaper and faster sequencing technology, the intensity peaks from ChIP-seq has also been shown to be useful for improving the results of motif discovery (Valouev et al., 2008). The spatial proximity with respect to the TSS (Transcriptional Start Site) has also been used as auxiliary information in (Thompson, 2003). Since DNA segments that are coiled around nucleosome have lower affinity for transcription factor and protein interactions, (Narlikar et al., 2007) has also used nucleosome positioning as *a priori* information to discern between the signal and noise. In Chapter 9, we will discuss the class of homology-aware motif finders that uses conservation information to improve the performance of motif-finding.

The large number of motif finders hints at the difficulty of the motif finding task and the need for a reliable statistical significance evaluation. Moreover, there has not been a general consensus as to which algorithms have the best performance, or for that matter which scoring metric is best for discovering motifs.

1.4 Why is statistical significance of motifs important?

For an experimentalist who is interested in discovering protein-DNA binding sites, he or she is interested in the *biological significance* of a motif discovered by a finder. Specifically, the experimentalist is interested in whether the protein-of-interest physically binds to the DNA at locations discovered by the algorithm. For

a transcription factor protein, in addition to the physical DNA-protein binding, an experimentalist would also be interested in whether a particular binding site would affect transcriptional regulation — since transcription factors can form a complex with other proteins and downregulate/upregulate transcription via indirect binding. In order to answer the question of biological significance, biological wet-lab assays (e.g. binding site knockout) that are expensive in time, labor, and cost are required. Moreover, it is an intractable and impossible computational problem.

Instead of attempting to address *biological significance*, we can reformulate the problem into a tractable computational problem and ask whether a motif is *statistically significant.* That is, is the predicted motif special (or statistically over-represented) when compared with a motif found in a random set of DNA sequences? One assumption behind using statistical significance is that a biologically significant binding site should be under selective pressure, and thus the binding sites should be well-conserved. In other words, a biological-relevant motif or pattern found in the original input should be more conserved than one found in a random set of DNA sequences. In fact, statistical significance evaluation assigns a probability p to the likelihood of a predicted motif being better than a motif found in a random set of DNA sequences. Statistical significance has a long history in design of experiments and data analysis; see (Durbin et al., 1999; Ewens and Grant, 2004) for a good review of the relevant techniques in this field.

Motif finders will typically report a motif given a set of input sequences, but an experimentalist must decide whether to invest significant resources in further exploration or verification of a reported motifs. Statistical significance is often the only information available to the users to help them make their decision. This is the reason why a reliable significance evaluation should be considered an essential

component of any finder. The computational evaluation of statistical significance for motif discovery is the central theme of this thesis, and it will be discussed in details in Chapters 5 and 6.

In addition, the actual motif finding task can be improved by an ensemble method[b] that is based on statistical significance evaluation[c]. The results of these ensemble algorithms will be discussed in Section 6.3.3.

[b] combining multiple motif finders to improve performance over its individual components

[c] specifically, by comparing p-values

Chapter 2

Motif scoring

2.1 Entropy score

One of the most widely used metrics to score a motif is the information content or (relative) entropy (Schneider et al., 1986; Hertz et al., 1990). Let n_{bj} be the number of occurrences of nucleotide b at the j-th position of the motif, i.e. $n_{bj} := \Theta_{bj} N$. Then the entropy of a given motif is:

$$I := \sum_{j=1}^{w} \sum_{b \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} n_{bj} \log \frac{\Theta_{bj}}{B_b} \tag{2.1}$$

where B_b is the background frequency of the nucleotide b. As we will see in Section 2.2, this score is also related to the likelihood ratio if we assume B is the model for the null hypothesis and Θ is the alternative hypothesis.

The background frequency B can be estimated from the input sequences by:

$$\hat{B}_b := \frac{\sum_{i=1}^{N} \sum_{j=1}^{l_i} \sum_{b \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} 1_{S_j^i = b}}{\sum_{i=1}^{N} l_i}$$

Note that the entropy is additive in this case, i.e. it can be decomposed to an entropy for each column of the motif:

$$I(j) := \sum_{b \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} n_{bj} \log \frac{\Theta_{bj}}{B_b}$$

Moreover if we naively assume that $B_b = \frac{1}{4}$ for all $b \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, then we get that

$$\begin{aligned} \frac{I(j)}{N} &= \sum_{b \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} \Theta_{bj} \log_2 \frac{\Theta_{bj}}{B_b} \\ &\approx 2 + \sum_{b \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} \Theta_{bj} \log_2 \Theta_{bj} \end{aligned}$$

which is exactly equivalent to (1.2) when we were discussing Weblogo. Similar to the example we gave when we were discussing Weblogo, note that $I(j) = 0$ if and only if $\Theta_{bj} = B_b$ for all $b \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$.

Finally the entropy score can be computed very quickly and it only requires Θ and B. It does not directly need the sequence-set S nor the location of motif instances Y_i . Given Θ and B, its time complexity is $O(w)$. In Section 3, we will discuss methods to optimize the entropy score given a set of sequences.

2.2 Likelihood ratio score (CLR)

The likelihood ratio score compares the likelihood between a null hypothesis versus an alternative hypothesis. In the case of motif discovery, the typical *null* is the hypothesis that the sequence-set S does not have a motif and the *alternative* is the hypothesis that S contains a motif. Assuming that data S is sampled from a generative model, the null is the hypothesis that S is generated completely by a background model B and the alternative is the hypothesis that S is generated by

motif model Θ at locations[a] Y and background model B. Hence the (complete) likelihood ratio (CLR) can be written as:

$$\begin{aligned} CLR(\Theta, Y \mid S) &:= \frac{P_{B,\Theta}(S \mid Y)}{P_B(S)} \\ &= \prod_{i=1}^{N} \frac{P_{B,\Theta}(S^i \mid Y_i)}{P_B(S^i)} \end{aligned} \tag{2.2}$$

which assumes each sequence S^i are independent. The background model B is often modeled using a higher-order Markov-chain. Assuming a k-th order Markov model, the denominator term for the null hypothesis would be

$$P_B(S^i) = P(S^i_1)\, P(S^i_2|S^i_1) \cdots P(S^i_k|S^i_{1:k-1}) \prod_{j=k+1}^{l_i} P(S^i_j|S^i_{j-k:j-1}) \tag{2.3}$$

The numerator term is more difficult to compute because it contains two models — motif model Θ and background model B. If we use a PWM to model motif, then we can write the alternative hypothesis term as:

$$\begin{aligned} P_{B,\Theta}(S^i \mid Y) &= P_B(S^i_{1:Y_i-1})\, P_\Theta(S^i_{Y_i:Y_i+w-1}|\Theta)\, P_B(S^i_{Y_i+w:l_i}) \\ &= P_B(S^i_{1:Y_i-1})\, (\prod_{m=1}^{w} \Theta_{S^i_{Y_i+m-1},m})\, P_B(S^i_{Y_i+w:l_i}) \end{aligned} \tag{2.4}$$

That is, the motif model Θ generates a motif instance of width w starting from location Y_i, while the rest of the sequence is generated by the background model B. Substituting with (2.4) and (2.3), we can simplify (2.2) to

$$\begin{aligned} CLR(\Theta, Y \mid S) &= \prod_{i=1}^{N} \frac{P_B(S^i_{1:Y_i-1})\, (\prod_{m=1}^{w} \Theta_{S^i_{Y_i+m-1},m})\, P_B(S^i_{Y_i+w:l_i})}{P_B(S^i)} \\ &= \prod_{i=1}^{N} \frac{\prod_{m=1}^{w} \Theta_{S^i_{Y_i+m-1},m}}{\prod_{j=Y_i}^{Y_i+w-1} P(S^i_j|S^i_{j-k:j-1})} \\ &\quad \times \frac{P_B(S^i_{Y_i+w:Y_i+w+k-1})}{\prod_{j=Y_i+w}^{Y_i+w+k-1} P(S^i_j|S^i_{j-k:j-1})} \\ &\approx \prod_{i=1}^{N} \prod_{m=1}^{w} \frac{\Theta_{S^i_{Y_i+m-1},m}}{P(S^i_{Y_i+m-1}|S^i_{Y_i+m-1-k:Y_i+m-2})} \end{aligned} \tag{2.5}$$

[a] under the OOPS (one occurrence per sequence model). See Section 2.4 for details

The last approximation of (2.5) comes from

$$\begin{aligned} P_B(S^i_{Y_i+w\,:\,Y_i+w+k-1}) &= P(S^i_{Y_i+w})\,P(S^i_{Y_i+w+1}|S^i_{Y_i+w}) \times \\ &\quad \cdots \times P(S^i_{Y_i+w+k-1}|S^i_{Y_i+w\,:\,Y_i+w+k-2}) \\ &\approx \prod_{j=Y_i+w}^{Y_i+w+k-1} P(S^i_j|S^i_{j-k\,:\,j-1}) \end{aligned}$$

Note that it is perfectly reasonable to simply use (2.4) as a scoring function because the background likelihood term (2.3) is constant given a fixed sequence-set S. However, computing the likelihood (2.4) is costly in running-time without precomputation. Assuming that the Markov chain probability $P_B(\cdot)$ is precomputed, the time complexity of ratio (2.5) is $O(Nw)$ while the straightforward computation of (2.4) takes $O(Nl_{max})$ where l_{max} is maximum sequence length. In addition, the implementation of (2.4) could easily encounter floating-point underflow for large l_{max} whereas the ratio (2.5) would circumvent that issue.

As we have mentioned in Section 2.1, the equation (2.1) is related to the likelihood ratio. Specifically it is equivalent to the logarithm of CLR for 0-th order Markov background model[b]. Let B_0 denote the 0-th order Markov background model, then the logarithm of CLR gives

$$\begin{aligned} \log CLR_{B_0}(\Theta, Y \,|\, S) &= \sum_{i=1}^{N}\sum_{m=1}^{w} \log \frac{\Theta_{S^i_{Y_i+m-1},\,m}}{P_{B_0}(S^i_{Y_i+m-1})} \\ &= \sum_{i=1}^{N}\sum_{m=1}^{w}\sum_{b\in\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} 1_{S_{Y_i+m-1}=b} \log \frac{\Theta_{bm}}{P_{B_0}(b)} \\ &= \sum_{m=1}^{w}\sum_{b\in\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} \left(\sum_{i=1}^{N} 1_{S_{Y_i+m-1}=b}\right) \log \frac{\Theta_{bm}}{P_{B_0}(b)} \\ &= \sum_{m=1}^{w}\sum_{b\in\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} n_{bj} \log \frac{\Theta_{bm}}{P_{B_0}(b)} \end{aligned}$$

[b] B is modeled by background frequencies.

One popular motif-finder implementation that uses Markov background model is BioProspector. It uses a slightly different variation from (2.5)

$$\prod_{i=1}^{N}\prod_{m=1}^{w}\frac{\Theta_{S^i_{Y_i+m-1},m}}{P_B(S^i_{Y_i:Y_i+w-1})}$$

which is only dependent on the nucleotides within the motif instances — ignoring the flanking nucleotides around the motif instance.

As we will discuss in Chapter 3, we know only of approximate algorithms to optimize the likelihood-ratio statistics[c] for Markov order $k \geq 2$. We can also only approximate its score distribution (see Chapter 5), but these approximations lay the grounds for a fairly reliable method to compute for statistical significance of motif discovery.

2.3 Incomplete likelihood ratio score (ILR)

The likelihood ratio score we discussed in Section 2.2 is a function of Θ and Y. But users are often more interested in discovering the motif profile Θ rather than the actual motif sites Y. Practically speaking, once a good motif profile has been discovered, users can scan the sequences for statistically significant motif instances (see SADMAMA (Keich et al., 2008)). Motif profiles are also more intuitive to visualize and understand and therefore the first thing that users are interested in after completing the motif-finding task. Subsequently we will now introduce a likelihood ratio statistics that does not directly depend on Y — *Incomplete*

[c]See (Nagarajan et al., 2006) for $k = 1$ case

Likelihood Ratio (ILR). We define the ILR as follows:

$$\begin{aligned} ILR(\Theta) &:= \frac{P_\Theta(S)}{P_B(S)} \\ &= \prod_{i=1}^{N} \frac{P_\Theta(S^i)}{P_B(S^i)} \\ &= \prod_{i=1}^{N} \left(\sum_{j=1}^{l_i-w+1} \frac{P(S^i_{j:j+w-1}|\Theta)}{l_i-w+1} \right) / P_B(S^i) \\ &= \prod_{i=1}^{N} \left(\sum_{j=1}^{l_i-w+1} \frac{\prod_{m=1}^{w} \Theta_{j+m-1,m}}{l_i-w+1} \right) / P_B(S^i) \end{aligned} \tag{2.6}$$

Similar to $CLR(\Theta, Y)$ statistics, the $ILR(\Theta)$ is the likelihood ratio between two competing hypotheses. The null hypothesis is that the data was entirely generated under the null model B. The alternative hypothesis is that the data was generated under the OOPS (one occurrence per sequence) model using the motif Θ and the background B. Unlike CLR score, the ILR scores a motif by taking into account all of the data in S, rather than only the data within a particular alignment. The ILR score in (2.6) assumes that there is no *a priori* positional bias and that the position for which a motif appears is uniformly distributed across a sequence. Furthermore, under the Bayesian framework with a uniform prior on Y_i , we can derive the $P_\Theta(S^i)$ term in (2.6) as

$$\begin{aligned} P_\Theta(S^i) &= \sum_{j=1}^{l_i-w+1} P(S^i, Y_i = j|\Theta) \\ &= \sum_{j=1}^{l_i-w+1} P(S^i|Y_i = j, \Theta)\, P(Y_i = j|\Theta) \\ &= \sum_{j=1}^{l_i-w+1} P(S^i|Y_i = j, \Theta)\, P(Y_i = j) \qquad \text{(no positional bias)} \\ &= \sum_{j=1}^{l_i-w+1} P(S^i_{j:j+w-1}|\Theta) \cdot \frac{1}{l_i-w+1} \end{aligned}$$

where $Y_i \sim U(1, l_i - w + 1)$. Although ILR is not a function of Y, recall that the motif profile Θ is typically estimated from Y by Equation (1.1). In fact, the estima-

tor $\hat{\Theta}$ in (1.1) is the maximum-likelihood estimator (MLE) of Θ given Y. However even when using the $\hat{\Theta}$ estimator, the ILR disregards positional information and averages across all possible motif sites for each sequence.

2.4 ZOOPS model

The scoring metrics we have discussed so far assume the OOPS (One Occurrence Per Sequence) model, i.e. the assumption that a motif instance appears *exactly* once per sequence. Since there may be noise in the input data and protein-DNA binding may be indirect (i.e. protein-of-interest is binding to a different protein that directly binds to the DNA), a motif instance often only occurs within a subset of the sequences. To circumvent the noise from polluting the motif signal, a scoring metric that assumes a ZOOPS (Zero or One Occurrence Per Sequence) model is often used. The following score uses a Bayesian prior on the percentage of sequences containing sites but uses a maximum likelihood (MLE) approach on the motif profile.

Our generative probabilistic ZOOPS model of the input set is defined as follows. Recall the given input to the model is: the number of sequences N, their lengths l_i, the (Markov) background model B and the motif modeled by a $4 \times w$ PWM Θ. We denote by p the probability that sequence S^i contains a site. We determine p by randomly drawing from a prior $\beta(a, b)$ distribution. In practice, we choose $a = b = \alpha N$ where α is a parameter that reflects the strength of your prior. Note that this choice indicates our prior belief that on average half the sequences should contain a site and it can readily be changed. This is because the expected number of sites is $NE(p) = Na/(a + b)$ which is $N/2$ if $a = b$.

We next draw N independent samples $\{Z_i\}_{i=1}^N$ from a Bernoulli(p) distribution. Each Z_i is the indicator function of the event 'sequence i contains a site'. Sequences i for which $Z_i = 0$ are not containing sites. Therefore, they are generated according to our background model B with probability[d] $P_B(S_i)$. Alternatively, if $Z_i = 1$ we first choose Y_i, the site[e] location for sequence i, uniformly from $\{1, 2, \ldots, l - w + 1\}$. We then generate the two background pieces $S^i_{[1\,:\,Y_i-1]}$ and $S^i_{[Y_i+w\,:\,l_i]}$, independently and according to the background model B. The site itself, $S^i_{[Y_i\,:\,Y_i+w-1]}$ is generated according to the product of multinomials parametrized by the PWM Θ: $\prod_{j=1}^w \Theta_{j,\,\alpha(j)}$, where $\alpha(j) := S^i_{Y_i+j-1}$. Below we sloppily refer to the product of these last three probabilities/likelihoods as $P_{B,\Theta}(S^i|Y_i)$.

The score we are trying to optimize is the model's joint likelihood:

$$\begin{aligned} P_{B,\Theta}(S, Z, Y) &= \int_0^1 \frac{p^{a-1}(1-p)^{a-1}}{\beta(a,a)} \cdot p^{\sum Z_i}(1-p)^{N-\sum Z_i} \\ &\quad \cdot \prod_{Z_i=1} \left[\frac{P_{B,\Theta}(S^i|Y_i)}{l_i - w + 1}\right] \cdot \prod_{Z_i=0} P_B(S^i)\, dp \\ &= \frac{\beta(\sum Z_i + a,\ N - \sum Z_i + a)}{\beta(a,a)} \\ &\quad \cdot \prod_{Z_i=1} \left[\frac{P_{B,\Theta}(S^i|Y_i)}{l_i - w + 1}\right] \cdot \prod_{Z_i=0} P_B(S^i) \end{aligned}$$

where $\beta(a,b) = \int_0^1 x^{a-1}(1-x)^{b-1}\, dx$ is the beta function. Specifically, we view Z and Y as missing parameters and we try to find

$$\text{argmax}_{\Theta,Z,Y}\, P_{B,\Theta}(S, Z, Y) = \text{argmax}_{Z,Y}\text{argmax}_{\Theta}\, P_{B,\Theta}(S, Z, Y)$$

Note that since the sites are generated according to a product of multinomials, maximizing over Θ given Z and Y, is the standard multinomial MLE (maximum likelihood estimation) deduced from the sites' letter counts.

[d] We adopt the common abuse of notations of failing to distinguish between the random variables and their actual values or instances.

[e] We ignore the issue of reverse complement sites in this discussion.

It is convenient to divide by the constant $\frac{1}{\beta(a,a)} \prod_i P_B(S^i)$ so that our target function simplifies to:

$$\Psi(Z, Y, \Theta|S) = \beta(\sum Z_i + a,\ N - \sum Z_i + a) \cdot \prod_{Z_i=1} \frac{P_{B,\Theta}(S^i|Y_i)}{P_B(S^i) \cdot (l_i - w + 1)} \quad (2.7)$$

Chapter 3

Motif finding algorithms

In Chapter 2, we described score metrics to evaluate motifs. But in order to discover new novel motifs within a sequence-set, we need algorithms that can search for a motif profile with the maximum score. As we will see in the next section, a brute force approach to search for such a motif profile is intractable. Moreover, there is no known efficient exact algorithm to perform this task for general likelihood ratio statistics[a], we therefore must resort to an approximate solution.

3.1 Gibbs sampler

Recall that we can quickly compute the CLR (complete likelihood ratio) score given the parameter-set (Θ, Y). Thus one can try to come up with an approximate

[a] e.g. ZOOPS model

maximization of the CLR score by sampling parameter-sets and then extract the one with the highest CLR. But how should we sample a parameter-set? Let's say we uniformly sample a random starting position Y_i for each sequence $i \in \{1, \ldots, N\}$ and then obtain $\hat{\Theta}_{MLE}$ from Y. It is easy to see that such a naive sampling scheme would be unfruitful because the the size of parameter space $\prod_{i=1}^{N} l_i$ is intractable for real-world input data.

The Gibbs sampling algorithm is an algorithm often used to sample from a joint probability distribution of multiple random variables (Geman and Geman, 1993). In this section, we expand its usage for optimizing the CLR, which was originally proposed by (Lawrence et al., 1993). By designing a Gibbs sampling algorithm that samples parameter-sets proportional to the CLR, we can optimize the CLR by saving the parameter-set that gives the highest CLR. Note that the advantage of Gibbs sampling over a naive uniform sampling approach is that the Gibbs sampling confines most of the search to regions with high CLR and is thus avoiding a uniform search over the entire parameter space.

The Gibbs-sampling motif finder begins each run by picking a random starting position in each sequence in the data set. The algorithm then iterates between two steps, commonly referred to as the predictive update step and the sampling step. The predictive update step computes a motif profile based on the current chosen set of starting positions. The sampling step in turn randomly selects new candidate starting positions with probability proportional to the likelihood ratio of the position given the current motif profile.

Formally, in the t-th iteration, we sample a set of new candidate starting positions Y_i^t for $1 \leq i \leq N$. Thus we iterate through each sequence i and the predictive update step would estimate the model Θ from the starting positions

$(Y_1^t, Y_2^t, \ldots, Y_{i-1}^t, Y_{i+1}^{t-1}, \ldots, Y_N^{t-1})$ by the rule

$$\widehat{\Theta_{bj}^{(t,i)}} = \frac{n_{bj}^{(t,i)} + \alpha_b}{N - 1 + \sum \alpha_b} \tag{3.1}$$

where n_{bj} is the count of letter b at the j-th position of the alignment and α_b is an *a priori* chosen pseudocount to avoid 0 probabilities. Note that $n_{bj}^{(t,i)}$ exclude contribution from sequence i so that the new motif profile is not biased toward Y_i^{t-1}. It depends on starting positions from the t-th iteration of sequences $\{1, \ldots, i-1\}$ and the $(t-1)$-th iteration of sequences $\{i+1, \ldots, N\}$

$$n_{bj}^{(t,i)} = \sum_{i'=1}^{i-1} \mathbf{1}(S^{i'}_{Y_{i'}^t + j - 1} = b) + \sum_{i'=i+1}^{N} \mathbf{1}(S^{i'}_{Y_{i'}^{t-1} + j - 1} = b)$$

where $\mathbf{1}(\cdot)$ is the indicator function.

In the sampling step, we sample a new candidate starting position Y_i^t given the model $\widehat{\Theta_{bj}^{(t,i)}}$ with probability[b]:

$$P(Y_i^t = y) \propto \frac{P_{B,\Theta}(S^i | Y_i = y)}{P_B(S^i)} \tag{3.2}$$

$$\propto \frac{\prod_{j=1}^{w} \Theta_{j,\, S^i_{y+j-1}}}{P_B(S^i_{y\,:\,y+w-1})} \tag{3.3}$$

which is proportional to the likelihood ratio of a motif instance starting at y generated by the motif profile $\widehat{\Theta_{bj}^{(t,i)}}$ versus by Markov background model B.

A well-known property of Gibbs-sampling algorithms is that they are guaranteed to sample the global maximum given sufficient time, but this may take an unacceptably long time to happen. Instead, when the objective function is apparently not making any headway, we can "restart" the sampling procedure by initializing a new, independent, Gibbs-sampling run using a new set of random starting positions.

[b]This is the sampling probability for the OOPS model. See Section 3.1.1 for the ZOOPS model.

3.1.1 Gibbs sampler under ZOOPS model

In Section 2.4, we discussed the CLR under the ZOOPS model. We will now show how to modify equation (3.2) to optimize the target function in equation (2.7). During the sampling step, we resample Y_i with probabilities proportional to:

$$P(Y_i = j) \propto \beta \left(\sum_{k \neq i} Z_k + a + 1,\ N - \sum_{k \neq i} Z_k + a - 1 \right) \cdot \frac{P_{B,\Theta}(S^i | Y_i = j)}{P_B(S^i) \cdot (l_i - w + 1)}$$

where $j \in \{1, 2, \ldots, l_i – w + 1\}$. We allow for $Z_i = 0$ which, following the convention mentioned in (Narlikar et al., 2007), we denote by $Y_i = 0$. Therefore, with the same proportionality constant as above:

$$P(Y_i = 0) \propto \beta(\sum_{k \neq i} Z_k + a,\ N - \sum_{k \neq i} Z_k + a)$$

As usual, we apply a plateau period condition [Lawrence et al., 1993] to stop a run and we use multiple runs, each with its own random starting locations. The configuration of Z and Y that maximizes $\Psi(Z, Y, \Theta | S)$ is reported together with its score.

3.2 Expectation-Maximization (EM)

The Expectation-Maximization (EM) algorithm Dempster et al. (1977) was formulated to compute the maximum likelihood estimator when the model depends on missing data. In this section we will apply this method (Bailey and Elkan, 1994) to optimizing the ILR score described in Section 2.3. The EM algorithm is also used within GibbsILR which will be described in Chapter 4. The EM algorithm iterates between two steps: the *E-step*

$$Q(\Theta | \Theta^{(t)}) := E_{Y \mid S, \Theta^{(t)}}[\log L(\Theta | S, Y)] \tag{3.4}$$

and the *M-step*

$$\Theta^{(t+1)} = \arg\max_{\Theta} Q(\Theta|\Theta^{(t)}) \tag{3.5}$$

where $L(\cdot)$ is the likelihood function, Y is the missing data, and $\Theta^{(t)}$ is the motif profile of the t-th iteration. It can be proven that the incomplete likelihood is non-decreasing during the iterative steps of EM, i.e. $L(\Theta^{(t+1)}|S) \geq L(\Theta^{(t)}|S)$. The proof of this result is beyond the scope of this thesis. Note that the EM algorithm converge only to a local maximum, thus multiple random "restart" is often applied in practice. We will now discuss how to apply the EM algorithm to our problem.

First the likelihood term $L(\Theta|S^i, Y_i)$ for sequence S^i can be written as

$$\begin{aligned} L(\Theta|S^i, Y_i = j) &= P(S^i, Y_i = j|\Theta) \\ &= P(S^i|Y_i = j, \Theta)\, P(Y_i = j) \\ &\propto \frac{\prod_{m=1}^{w} \Theta_{S_{j+m-1},m}}{(l_i - w + 1)\, P_B(S^i_{j:j+w-1})} \end{aligned}$$

We initialize the EM algorithm with a random motif profile $\Theta^{(0)}$. Then we would locally improve the motif profile by the E-step and M-step, i.e. we estimate $\Theta^{(t+1)}$ from $\Theta^{(t)}$. The equation in (3.4) of E-step is rewritten as follow:

$$\begin{aligned} Q(\Theta|\Theta^{(t)}) &= E_{Y\,|\,S,\Theta^{(t)}}[\log L(\Theta|S, Y)] \\ &= \sum_{i=1}^{N} E_{Y\,|\,S,\Theta^{(t)}}[\log L(\Theta|S^i, Y_i)] \\ &= \sum_{i=1}^{N} \sum_{j=1}^{l_i-w+1} P(Y_i = j\,|\,S^i, \Theta^{(t)})\, \log L(\Theta|S^i, Y_i = j) \\ &= \sum_{i=1}^{N} \sum_{j=1}^{l_i-w+1} P(Y_i = j\,|\,S^i, \Theta^{(t)})\, \log \prod_{m=1}^{w} \Theta_{S_{j+m-1},m} + C \\ &= \sum_{b\in\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}} \sum_{m=1}^{w} \sum_{i=1}^{N} \sum_{j=1}^{l_i-w+1} P(Y_i = j\,|\,S^i, \Theta^{(t)}) \\ &\quad \times \mathbf{1}(S_{j+m-1} = b)\, \log \Theta_{bm} + C \end{aligned} \tag{3.6}$$

where C does not depend[c] on Θ. To maximize $Q(\Theta|\Theta^{(t)})$ in the M-step, observe that (3.6) is the logarithm of the multinomial distribution with parameters Θ_{bm}. Since the MLE of multinomial distribution is well-known, we have that the M-step is given by:

$$\Theta_{bm}^{(t+1)} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{l_i-w+1} P(Y_i = j \mid S^i, \Theta^{(t)})\,\mathbf{1}(S_{j+m-1} = b)}{\sum_{b'\in\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}}\sum_{i=1}^{N}\sum_{j=1}^{l_i-w+1} P(Y_i = j \mid S^i, \Theta^{(t)})\,\mathbf{1}(S_{j+m-1} = b')}$$

where

$$P(Y_i = j \mid S^i, \Theta^{(t)}) = \frac{\left[\prod_{m=1}^{w}\Theta_{S_{j+m-1},m}\right]/P_B(S^i_{j:j+w-1})}{\sum_{j'=1}^{l_i-w+1}\left[\prod_{m=1}^{w}\Theta_{S_{j'+m-1},m}\right]/P_B(S^i_{j':j'+w-1})}$$

[c]We disregard C because it does not affect $\Theta^{(t+1)}$ within maximization of M-step.

Chapter 4

GibbsILR

(Keich and Pevzner, 2002a) define a twilight zone search as one in which there is a non-negligible probability that a maximally scoring random motif would have a higher score than motifs that overlap the ‘‘real’’ motif (in the model considered there, a ‘‘real’’ motif is implanted into randomly generated background sequences). In such cases, even if one had access to a hypothetically ideal finder that was guaranteed to return the highest scoring alignment in the dataset, the motif might remain unfound. Improving motif finding tools to be effective into the twilight zone is not merely a theoretical exercise: a biologist searching for regulatory motifs in DNA sequences would generally prefer to choose longer rather than shorter regions in order to avoid missing regulatory elements that are far away from the transcription start site of a gene. The longer the input sequences are, the more likely they are to contain high scoring random motifs, pushing the biologically valid motifs into the twilight zone.

This chapter discusses using Incomplete Likelihood Ratio (ILR)[a] and our GibbsILR motif finder for improving motif finding within such twilight zone search. We begin by showing that comparison of entropy scores across different motif finders often leads to inconsistent results regarding the identification of the implanted motif. On the other hand, we observe that ILR is a significantly better classifier when it comes to predicting overlap with implanted motifs for twilight zone searches. Thus this motivates GibbsILR, a new variant of the Gibbs sampler that attempts to maximize the ILR rather than the entropy score.

4.1 Fallacy of entropy score

A good scoring function should separate as much as possible real motifs or, in the context of our model, alignments that have overlap with the implant, from purely random ones. The entropy score is the one chosen by popular motif finders such as MEME (Bailey and Elkan, 1995), CONSENSUS (Hertz and Stormo, 1999) and Gibbs Sampler (Lawrence et al., 1993; Neuwald et al., 1995; Hughes et al., 2000). The latter two specifically try to optimize this scoring function, while MEME uses it only to rank and analyze the significance of its output. It is thus tempting to assume that if we run, for example, both CONSENSUS and Gibbs and take the higher scoring motif we would do better than if we ran each one of them separately. Amazingly, this might not be the case, especially in twilight zone searches. To answer this, we first designed the following experiment containing 400 data sets with implanted twilight zone motifs (see COMBO experiments in Section 4.4 for details). In particular, each randomly generated data set contained a deliberately implanted profile motif in such a way that for a non-trivial percentage

[a] See Section 2.3

of datasets, the motif finders we considered would pick motifs that would not overlap the implants. In this COMBO experiment, we find that in 380 of the 400 datasets CONSENSUS finds a motif with higher entropy score than Gibbs, yet Gibbs reports more motifs that have $\geq 30\%$ overlap with the true implant (290 of the sets for Gibbs compared to 208 for CONSENSUS). Comparing the entropy score from different motif finders is thus not an apples to apples comparison as one would expect—somehow it matters how the entropy is maximized. This led us to ask if other scoring functions would possibly capture better the nature of real (implanted) twilight zone motifs.

4.2 GibbsILR algorithm

As we will show in the next Section, we found that for twilight zone searches ILR is a significantly better classifier than the entropy score for identifying motifs that overlap the implant. Hence we designed a new finder that tries to optimize the ILR: GibbsILR is based on the Gibbs-sampling technique described by (Lawrence et al., 1993). Unlike previous Gibbs-sampling motif finders (see Section 3.1), GibbsILR runs an EM (Expectation-Maximization) algorithm that locally optimizes ILR on the final motif of each Gibbs-sampling run. GibbsILR then produces a motif that exhibits locally optimized ILR score by taking the highest ILR-scoring motif among all of the final motifs derived from the EM step. Finally, for each sequence in the dataset S, the motif instance corresponds to the position with the highest likelihood ratio with respect to the highest ILR-scoring motif profile.

4.3 Comparing entropy and ILR score

The first group of results is based on extensive tests of the performance of six profile-based motif finders on synthetic data. Each of these randomly generated datasets contained a deliberately implanted profile motif (see Section 4.4 for experimental methods). The output of each of the finders we considered (CONSENSUS, Gibbs, GibbsILR, GLAM, ProfileBranching, and MEME) was post-processed to yield both the entropy and ILR scores of the finder's top reported alignment. We then asked which of these two scores is a better predictor of overlap with the implant (which is a surrogate for a real motif).

We compare the entropy and ILR score by measuring the area under the ROC curve (Swets, 1988) or discrimination, for each finder under the two scoring functions. We classify a set of motif sites as negative if the overlap score[b] is below 0.1; otherwise, we classify it as positive. Intuitively, given a random pair of positive and negative set of profile sites, the aROC tells us the probability of the test correctly identifying the pair's classifications. The tests (see Table 4.1) using ILR score have better discrimination than the tests using entropy score. The reader should note that it is however unfair to compare the performance of the finders using aROC, because the number of negatives and positives differ across the finders. For example, GibbsILR has lower discrimination than MEME for both entropy and ILR in COMBO, but GibbsILR has 324 positives to discriminate whereas MEME has only 70 positives.

Similarly we can ask how many true positives (TP) are in the test set if we are willing to accept exactly 10 false positives (FP) (see Table 4.1). If we would like to design a classifier that only accepts 10 FPs, this analysis shows that the

[b]overlap score is defined in Section 4.4

Table 4.1: **aROC and only accepting 10 FPs.** The column > 10% contains the number of datasets that score above the 0.1 overlap threshold. The column TPs contains the number of true-positives in a test if it is willing to accept ≤ 10 FPs.

			entropy		ILR	
Experiment	**Finders**	**> 10%**	**aROC**	**TPs**	**aROC**	**TPs**
COMBO	CONSENSUS	223	0.88	154	0.93	169
	Gibbs_ss	302	0.88	208	0.91	231
	GibbsILR	324	0.85	254	0.90	258
	GLAM	170	0.90	117	0.94	127
	MEME	70	0.90	43	0.92	48
	ProfileBranching	222	0.95	183	0.96	190
FIFTY	CONSENSUS	27	0.73	5	0.85	13
	Gibbs_ss	87	0.94	70	0.96	76
	GibbsILR	186	0.96	171	0.96	171
	GLAM	116	0.94	91	0.89	84
	MEME	4	0.64	0	0.73	0
	ProfileBranching	8	0.60	1	0.72	1

combination of ILR score and GibbsILR would give us the highest number of TPs.

We next combine five motif finders: CONSENSUS, Gibbs_ss, GLAM, MEME, and ProfileBranching by choosing the set of motif sites from the finder with highest ILR. Likewise, we employ the same technique with the entropy. We found that the ILR variant of the combined-finder can perform better than any of its individual finders alone. In the COMBO experiment, the ILR variant found the implants in 311 datasets (i.e. overlap score greater than 0.1), whereas its best individual finder, which is Gibbs_ss, found the implants in 302 datasets. In the same experiment, the entropy variant found the implants in 291 datasets, which is worse than its best individual finder. For a different approach to combining the output of multiple motif finding algorithms, see (Hu et al., 2005).

As an additional source of evidence for the utility of the ILR score we generated synthetic data sets implanted with motifs that were verifiably in the (entropy score) twilight zone. Then a branch and bound algorithm was used to find the motif with the optimal entropy score and the ILR score of that motif. Then, based on the results from 1000 such runs we asked the following question: which of the scores, entropy or ILR is a better predictor of overlap with the implanted motif? For the twilight zone data sets that we tested, ILR is consistently better than the entropy score as a predictor of overlap (as measured by the aROC score, with overlap being defined as an overlap score greater than 0.1). As a specific example, for $N = 14$, $L = 80$ and `SHORT` (see Table 4.2), the entropy score has an aROC score of 0.52 as compared to 0.60 for the ILR score. In practical terms, for a threshold that allows 50 false positives, the ILR score gives 143 true positives as opposed to 101 for the entropy score. Interestingly, in this example, while the ILR score has a positive Spearman correlation, the entropy score has a statistically significant negative correlation with the overlap score (Spearman correlation p-value of $5.2 \cdot 10^{-4}$). Recall that the detected motif was optimized for the entropy, rather than the ILR.

Finally while not an objective demonstration of the advantage of ILR, GibbsILR did show improvement in our experiments over the other five finders we tested. Figure 4.1 and 4.2 show the overlap distribution for the various finders. For example, the bars at 0.1 are the number of datasets that a particular motif finder found with overlap score between 0.1 and 0.2. GibbsILR finds the most datasets above 0.1 overlap score for both experiments. In the case of `FIFTY` (See Figure 4.2), GibbsILR is significantly better. Note that we tried to equalize the running time of all the algorithms in the benchmark as described in the Methods section below.

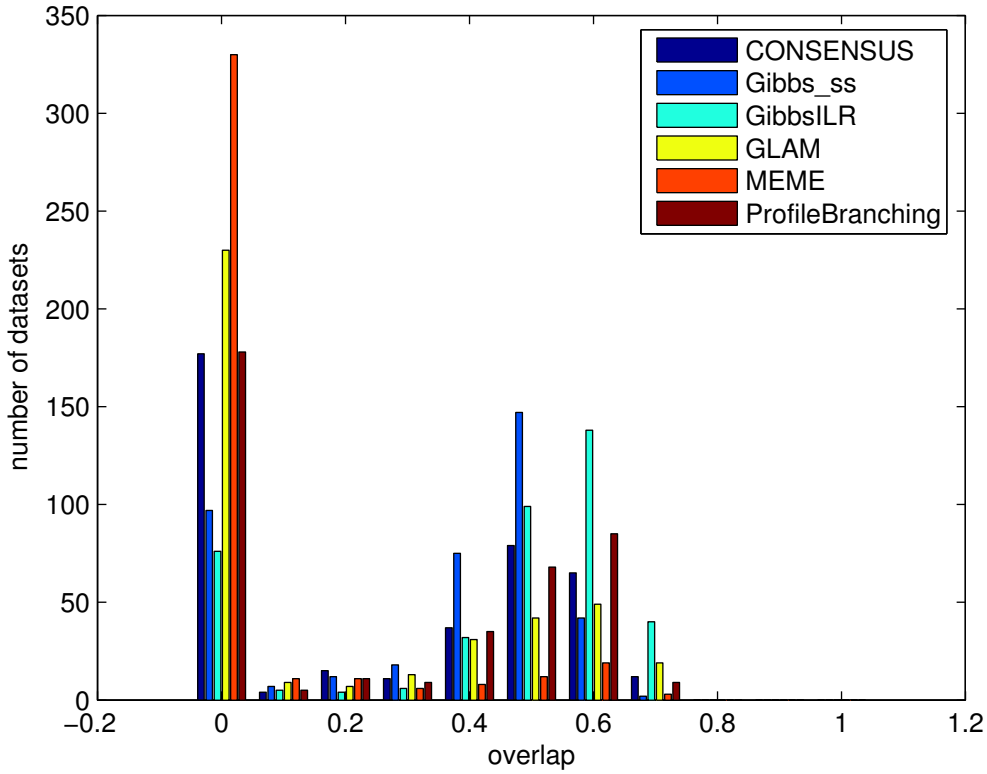


Figure 4.1: **COMBO experiment**: Histogram of the number of datasets as a function of the amount of overlap with the implanted motif.

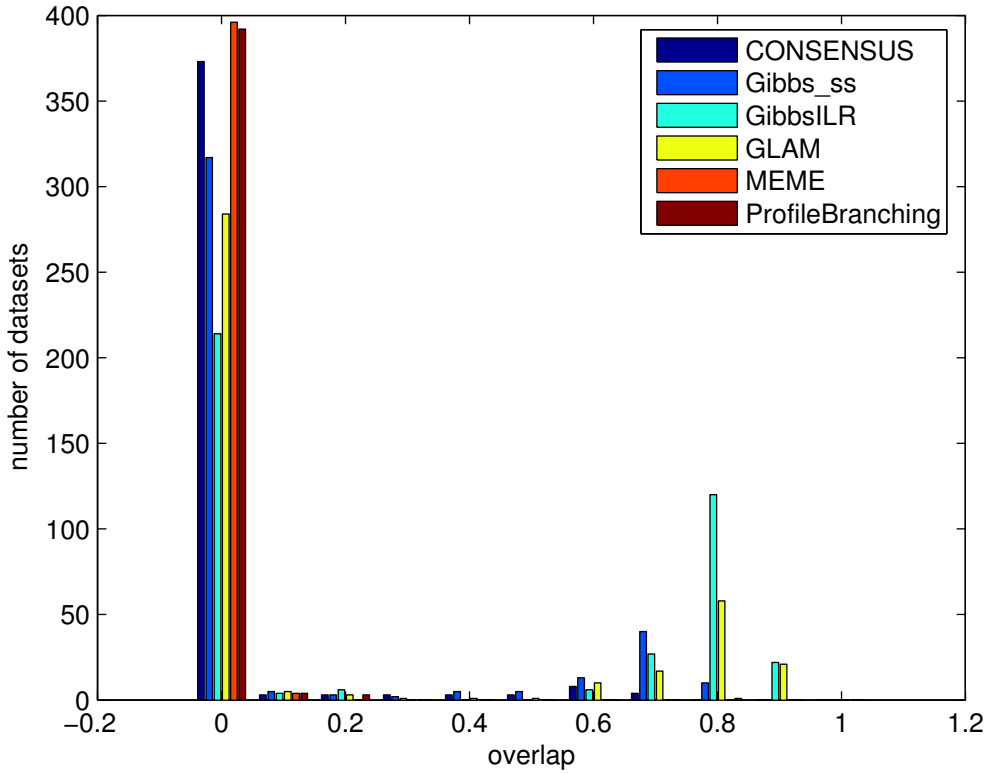


Figure 4.2: **FIFTY experiment**: Histogram of the number of datasets as a function of the amount of overlap with the implanted motif.

4.4 Methods for GibbsILR experiments

To test the efficacy of any given motif finding algorithm, N independent sequences of length m were sampled by choosing symbols at random from the four letter DNA alphabet corresponding to an iid model for the background frequency. A position was chosen uniformly at random from each sequence and an instance of a profile Θ, generated as described below, was inserted in that position. Thus, the total length of each sequence is $L = m + w$ where w is the length of the motif. A profile is represented as a position weight matrix, a $4 \times w$ array of numbers where Θ_{ij} denotes the frequency of letter i in column j in all aligned instances of Θ. Since we wanted to have control over the implanted motifs the instance were essentially generated by permuting the columns of the alignment. Each column of the alignment matched the corresponding column of the profile up to discretizing effects.

The parameters N and L were chosen such that the motif finders we considered would have a non-trivial percentage of failures (i.e. datasets where they pick motifs with no overlap with the implants). As we allowed our finders to run for a fairly generous amount of time there is reason to suspect that at least some of those failures can be attributed to twilight zone searches (Keich and Pevzner, 2002b), in which random alignments with no overlap with the implants score as high as the best motif that overlaps the implant.

Two of the experiments that we report here were generated according to the following rules:

1. `COMBO`: The motif in this experiment has length 13 with two degenerate columns (6 and 8) as seen in Table 4.2. Each dataset has 40 sequences

of length 1485 + 13.

2. `FIFTY`: Each column in the motif consists only of two equally probable nucleotides. Each dataset has 40 sequences of length 1485 + 13.

In each experiment, 400 datasets were generated for a given profile, and various motif finding algorithms were run with parameter settings that allowed each motif finder to take from 8–10 minutes to place all motif finders on an equal footing. However, the MEME motif finder does not employ any parameters that allow the control of running time (MEME generally runs in much less than 8 minutes on each data set), so the generally poor performance of MEME compared to the other motif finders is not a reflection of MEME employing a bad algorithm but a reflection of a design decision to place a strict limit on the total amount of time MEME takes. The motif finders used in this study consisted of MEME (Bailey and Elkan, 1995) (`-mod oops - nmotifs 1 -w 13 -dna -text -maxsize 1000000`), the Gibbs Sampler run in Site Sampler (''Gibbs_ss'') of (Lawrence et al., 1993) (`13 -d -n -t280 -L200`), Gibbs altered to use the ILR scoring function (''GibbsILR'', `13 -t 250 -L 200 -p 0.05`), GLAM (Frith et al., 2004) (`-n50000 -r10 -1 -z -a13 -b13`), CONSENSUS (Hertz and Stormo, 1999) (`-L 13 -c0 -q 3000`), and ProfileBranching (Price et al., 2003) (`-l 13 -verbose`). We note that Gibbs_ss is our version of the original algorithm optimized for site sampling mode, resulting in a three-fold improvement in running time. For this reason, the results of Gibbs_ss are better than the results of the original algorithm for a fixed running time. All experiments were run under Red Hat Enterprise Linux 4 on a cluster with nodes that have AMD 248 2Ghz 64-bit processors with 2GB RAM and 1GB swap.

An estimate of overlap for each data set and for each motif finder was computed in the following manner: Let a_n be the position of the implanted motif instance in

Table 4.2: The position weight matrices used in these experiments

	COMBO				FIFTY				SHORT			
Pos.	A	C	G	T	A	C	G	T	A	C	G	T
1	0.95	0.00	0.00	0.05	0.50	0.00	0.00	0.50	0.95	0.00	0.05	0.00
2	0.00	0.50	0.50	0.00	0.00	0.50	0.50	0.00	0.00	0.05	0.95	0.00
3	0.70	0.10	0.10	0.10	0.50	0.50	0.00	0.00	0.29	0.29	0.21	0.21
4	0.00	0.70	0.30	0.00	0.50	0.00	0.50	0.00	0.00	0.00	0.50	0.50
5	0.50	0.00	0.00	0.50	0.50	0.50	0.00	0.00	0.00	0.00	0.05	0.95
6	0.25	0.25	0.25	0.25	0.00	0.50	0.50	0.00				
7	0.95	0.00	0.00	0.05	0.00	0.50	0.00	0.50				
8	0.25	0.25	0.25	0.25	0.00	0.50	0.00	0.50				
9	0.70	0.10	0.10	0.10	0.50	0.00	0.50	0.00				
10	0.00	0.50	0.00	0.50	0.00	0.50	0.50	0.00				
11	0.00	0.70	0.00	0.30	0.50	0.50	0.00	0.00				
12	0.70	0.10	0.10	0.10	0.00	0.50	0.50	0.00				
13	0.00	0.50	0.50	0.00	0.00	0.50	0.00	0.50				

S^n, and let $\widehat{a_n}$ be the position of the motif reported by a motif finder. We define the *overlap* of a motif finder's prediction as:

$$\max_{|i|<\frac{w}{2}} \left\{ \frac{w - |i|}{w} \cdot \frac{|\{n : a_n = \widehat{a_n} + i\}|}{N} \right\}$$

All ILR scores in this chapter were computed using a uniform pseudocount of 0.05.

Chapter 5

Motif score distribution

In most applications of a motif finder, the user must decide whether or not a motif reported from a motif finder warrants further biological investigation based on its statistical significance. This chapter deals with the significance analysis of the ubiquitous entropy (or generally CLR) score. We begin by showing that the common practice of using the E-value of the entropy score (defined below) to evaluate the significance of an alignment reported by a motif finder can lead to undesirable results in twilight zone searches. We then discuss two additional intuitively motivated measurements of statistical significance and some pitfalls in their application to motif finding.

5.1 Are motif-finders psychic? Conundrum of E-value

One of the key measurements in determining if a motif finder has identified an important motif is the E-value of the entropy score. Introduced originally in this context as the "expected frequency" (Hertz and Stormo, 1999), the E-value is the expected number of random alignments of the same dimension that would exhibit an entropy score that is at least as high as the score of the given alignment. When the E-value is high, one can have little confidence in the motif prediction, and conversely when the E-value is low, one can have more confidence in the prediction. It is computed by multiplying the number of possible alignments by the p-value of the alignment. The latter is defined as the probability that a single given random alignment would have an entropy score $\geq$ the observed alignment score. Assuming the customary iid (independent identically distributed) random model the p-value can be computed accurately using techniques described in (Nagarajan et al., 2005).

Recall the COMBO experiment that we have described in Section 4.4, it consists of 400 randomly generated data sets with implanted twilight zone motifs. Thus it is not surprising that the E-values of their implanted motifs are relatively high. However, with a median E-value of 8×10^{15} it seems this problem is way beyond the twilight zone. Indeed, one would suspect that in this case even the ideal finder would not be able to pick out an alignment with significant overlap to the implanted motif from the large number of background alignments with better entropy score. Rather startlingly, exactly the opposite is true: of 400 data sets, the Gibbs sampler (Lawrence et al., 1993) found an alignment overlapping more

than 30% of the implanted sites in 288 cases[a]. It is important to note that these data sets are constructed exactly according to the model used in computing the E-values, thus we can safely assume the E-value is quite accurate (Nagarajan et al., 2005).

How can our motif finders be so lucky that they pick a "real" motif out of such a huge haystack? A partial answer to this riddle is obtained by noting that when a motif is implanted into a set of long sequences, there is a good chance that a random string in one of the sequences will slightly improve the entropy score. Of the 288 data sets for which the Gibbs sampler found an overlapping alignment (above the 30% threshold), the median E-value of the reported motif was 8.7×10^{11} or 4 orders of magnitude better than the initial motif. Still, it is a very impressive haystack and a more complete answer probably lies in what we do not see: how many alignments that overlap with our implant have a score as good as the one found? These high scoring ''satellite'' alignments define some ''domain of attraction'' for a motif that is difficult to characterize analytically. Presumably, its size has to be of the order of the E-values as sampling optimization procedures such as Gibbs somehow find it. We remark that characterizing this domain of attraction is a potential way to describe the twilight zone of a profile-based motif.

Whatever the explanation is, it is clear that the E-value offers little benefit in analyzing the significance of twilight zone search. We next explore alternative approaches to this problem.

[a]See Section 4.4 for the parameters setting.

5.2 Statistical significance based on (null) distribution of finder's score

One alternate measure of significance suggested by (Hertz and Stormo, 1999) is that of the "overall p-value" — or $OPV(s)$ — of an entropy score s. It is defined as the probability that a random sample of the same size as the input set will contain an alignment of the same dimensions that scores at least as high as s. While this statistic is intuitively appealing, its use faces two hurdles. On the one hand, at present it is all but impossible to calculate $OPV(s)$ for moderately large datasets: even generating an empirical estimate of the OPV would necessarily require the ability to reliably find the highest scoring alignment in any given sample, which cannot be guaranteed for realistic problem sizes. On the other hand, even if an accurate method for calculating $OPV(s)$ were known, the evidence presented next suggests that this significance measure would impose too high a barrier on the entropy score for functional motifs to be distinguishable from noise.

The value of $OPV(s)$ may be conservatively estimated by the probability that at least one of several motif finders would find an alignment of score $\geq s$ in the random data. The point is the latter is amenable to Monte Carlo estimation. Using 1600 randomly generated datasets with no motif implanted we obtain an empirical estimate of the 0.95 quantile of the latter distribution; this is the minimal value s_0 such that for 95% of the datasets all our finders report a top alignment of score $\leq s_0$. We then use s_0 as an empirically derived conservative estimate of the threshold s_1 such that $OPV(S_1) = 0.95$. That is, 95% of the top scoring noise alignments have entropy less than or equal to s_1 and $s_1 \geq s_0$ with high probability. When this derived 5% significance level was applied as a threshold for significance

of the 400 data sets in the COMBO experiment, nearly 90% of the correct runs of the Gibbs sampler (i.e. those runs that overlapped the implanted motif by more than 30%) were classified as noise. Since s_0 the conservatively estimated 0.95 quantile is very likely to be less than the true quantile s_1, this should become more pronounced with better approximations of $OPV(s)$ suggesting it is also too conservative.

One can see that $1 - OPV(s)$ is the distribution function of the ideal motif finder. This raises the natural extension of using a finder-specific OPV: $1 - F_f(s)$ where F_f is the null distribution of the score of the optimal alignment detected by the particular finder. That is, we ask for the probability that the finder will find an alignment scoring $\geq s$ in a random dataset (of the same dimensions). Again, we can estimate the quantiles of this distribution function through a Monte Carlo generated empirical distribution. In this case we found that the .95 quantile threshold of Gibbs, estimated from 1600 datasets, yields 13 false positives (FP) and 228 true positives (TP) when applied to the same 400 data sets of the COMBO experiment[b].

While the empirical distribution can be extremely useful in analyzing the significance of a motif finder's output, generating it *a priori* is typically impossible due to the large number of combinations of parameters. Similarly, generating even a rough estimate of a 0.95 quantile per problem instance is impractical as it would require at least 100 additional runs of the motif finder on a dataset of the same size as the input.

However, if we can characterize the distribution as belonging to some paramet-

[b]We expect $5\% \cdot 400 = 20$ false positives and see only 13 is reasonable since some of those random datasets containing high-scoring alignments are masked by higher-scoring motifs that overlap the implant.

ric family of distributions, we might do better to estimate the parameters of the distribution rather than directly estimating the quantiles of the distribution. The (limiting) distribution of a maximal ungapped pairwise alignment between two sequences is a Gumbel Extreme Value Distribution (EVD) (Karlin and Altschul, 1990); the same distribution is encountered empirically in the gapped case and it is presumed to underlie the distribution of scores when local multiple alignments are scored according to a presumed phylogeny (Prakash and Tompa, 2005) and in (Frith et al., 2004), which specifically discusses motif finding. Oddly, the empirical null distribution of the reported entropy score for several motif finders exhibited a better fit to a 3-Gamma distribution[c] than to the intuitively more appealing Gumbel distribution (see Figure 5.1 and Figure 5.2 for an example involving Gibbs).[d]

We demonstrated above that the OPV or equivalently the distribution function of the ideal finder seems too conservative for estimating the significance of a motif finder's output. Nonetheless it is useful in delineating the twilight zone, which in turn is important for understanding to what extent existing tools might be theoretically improved upon. Indeed, by comparing the empirical distribution of a motif finder with that of the ideal one for a given set of parameters, we can assess the efficiency of the finder for these parameters. It is thus interesting to determine whether this distribution can be approximated by a parametric family. As above we find the surprising result that a 3-Gamma distribution gives a better fit to the empirical distribution than a Gumbel distribution. One might expect that the result of maximizing over all possible alignments would naturally result in an EVD

[c]The distribution function of a 3-parameters Gamma with $\theta = (a, b, \mu)$ is a given by $F_\theta(s) = F_{\Gamma(a,b)}(s-\mu)$where $F_{\Gamma(a,b)}$ is the Gamma distribution with it usual shape and scale parameters and μ is the location parameter.

[d]To fit a 3-Gamma distribution for each shift we find the likelihood of the shifted data by applying a standard maximum likelihood gamma fit to it, and then use a simple one dimensional search of the shift that yields the highest likelihood.

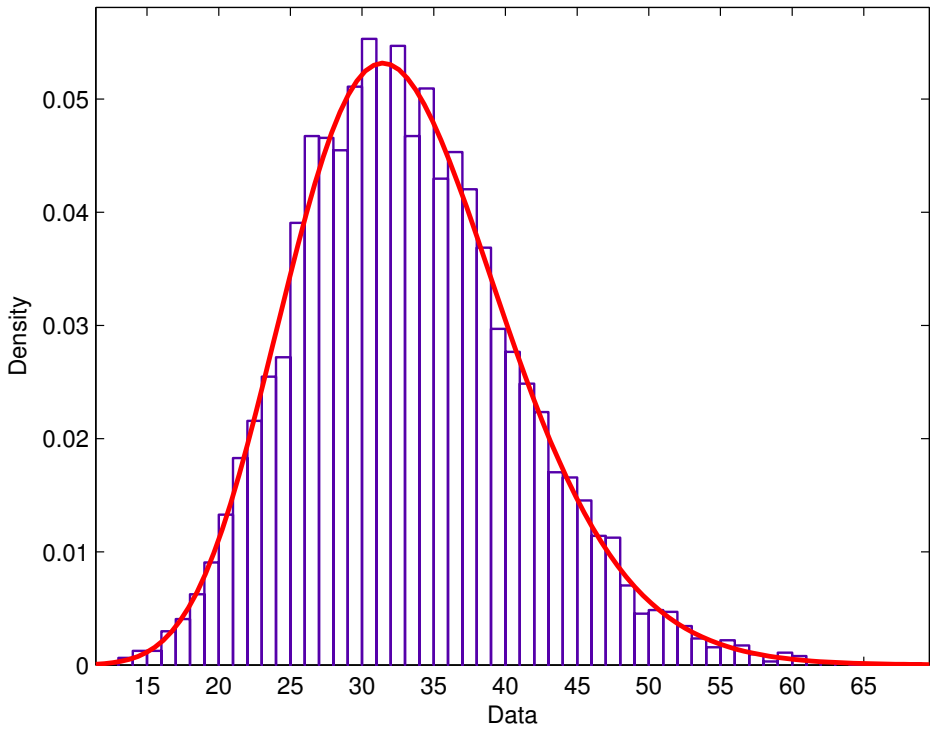


Figure 5.1: 3-Gamma fit to 6400 runs of Gibbs with parameters `13 -t100 -L100` on 40 random sequences of length 750, uniformly distributed with no implanted motif.

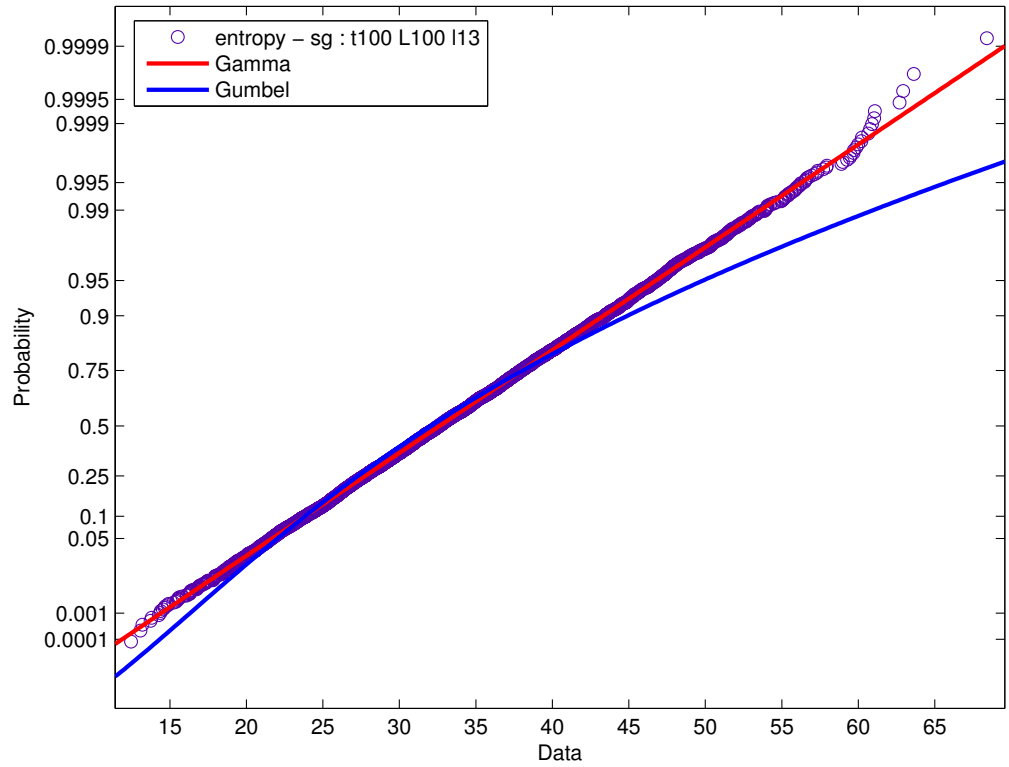


Figure 5.2: The probability plot of the fit of a 3-Gamma distribution and of a Gumbel Extreme Value Distribution to the data collected in Figure 5.1.

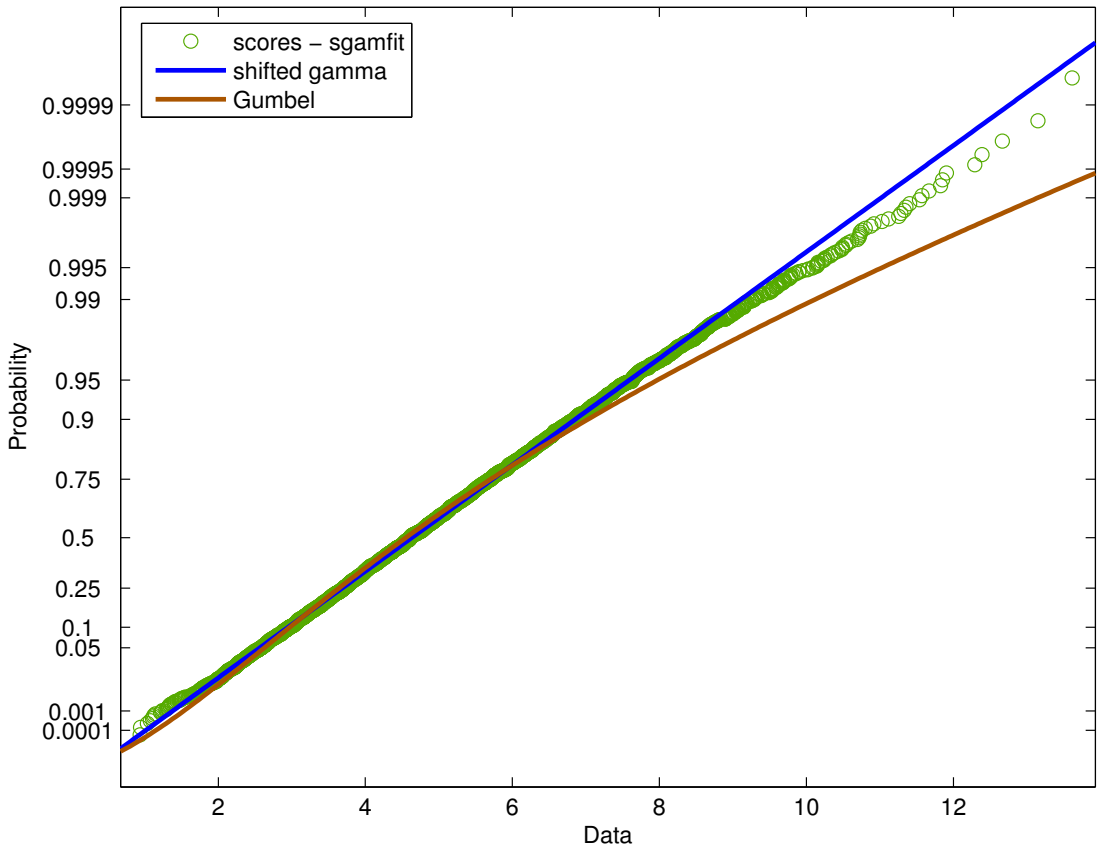


Figure 5.3: The probability plot of a fit of the OPV distribution to a 3-Gamma distribution and to an EVD distribution; the OPV distribution was generated from the output of the ideal motif finder (searching for motifs of width 7) run on 10000 datasets composed of 10 uniformly distributed sequences with length 100.

but according to our observations this is not the case[e] (see Figure 5.3). One reason is that the high scoring alignments are heavily dependent, an observation made by (Frith et al., 2004) when trying to explain the less-than-perfect fit they got to a Gumbel distribution.

We will see in the next Chapter 6 how we can use the OPV distribution and its 3-Gamma fit to come up with a reliable technique for significance evaluation. Additionally, we will show its reliability and performance on real motif data from *Saccharomyces cerevisiae* genome.

[e]However, the fit to GEV is as good as the one to 3-Gamma.

Chapter 6

Motif significance on real-data

In Chapter 5, we argued that the finder's null distribution is well suited for estimating the significance of a finder's output. In this chapter, we introduce a reliable method to estimate "confidence" p-values from a small sample of the empirical null distribution of a motif finder's results. We then naturally extend our confidence p-value approach to incorporate local base composition information. Furthermore, we demonstrate the ability of our local composition aware significance evaluation to reliably predict significant motifs in real biological setting.

6.1 Small-sample parametric approach

Recall that the finder's null distribution is defined as the distribution of the score of the finder on a randomly drawn set, generated for example by resampling a large genomic file. Note that this distribution varies not only with the null model that

generates the dataset (including the set's dimensions), but also with the parameters of the finder (e.g., width). Since there are typically infinitely many combinations of these problem-parameters (finder and dataset) it is impossible to precompute this distribution.

For any specific set of problem-parameters we can approximate the finder's null distribution with an empirical null distribution, but such a non-parametric approach to reliably estimate small p-values[a] is typically forbiddingly expensive. at a significant cost. However, if we know that the finder's null distribution can be well approximated by some parametric family, then we only need to estimate these parameters based on a *small sample* of the empirical null distribution (see Algorithm 6.1).

While the normal distribution is often used in this context (Harbison et al., 2004; Liu et al., 1995; MacIsaac et al., 2006; Narlikar et al., 2007), we find that it consistently offers a relatively poor approximation to the finder's null distribution. In particular, using the normal approximation tends to inflate the significance of high scores which are the ones we are interested in (see Figure 6.2). Instead, as we have discussed in Chapter 5, we find that the 3-parameter Gamma (Johnson et al., 1994), or 3-Gamma for short, appears to fit very well to the empirical null distribution for many combinations of motif finders and null models including the biologically realistic, genomic resampling (see Figure 6.2).

Technically, suppose we want to estimate the p-value of the observed score s, denoted by $p(s)$, assuming a 3-Gamma approximation. We can generate a small sample $X = (X_1, \ldots, X_n)$ from the finder's null distribution and find the 3-Gamma

[a] we often need to estimate small p-values to correct for multiple hypotheses (e.g. Harbison dataset has over 300 experiments (Harbison et al., 2004))

Algorithm 6.1 Motif p-value evaluation

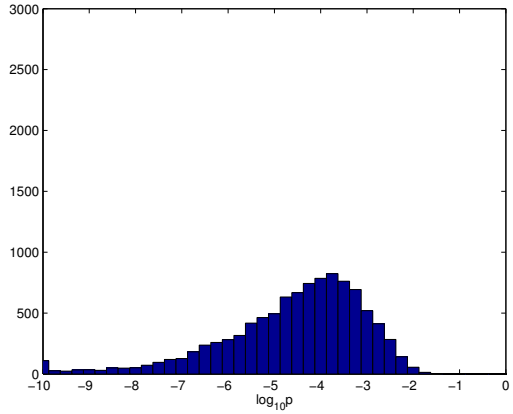
- Precompute a parametric distribution family for motif-finder
    1. Run motif-finder on a large set (e.g. 10K samples) of random sequence-sets
    2. Find a parametric family that is a good fit for the scores (e.g. 3-Gamma, Gaussian, GEV distributions)
- Given motif input data (e.g. ChIP-chip data)
    1. Generate a small set (e.g. 30 samples) of random sequence-sets mimicking the original input data
    2. Run motif-finder on the input set and the small random set
    3. Estimate parameters of the *a priori* parametric distribution using scores from the random set
    4. Perform statistical significance p-value evaluation based on parametric fit from Step 3 and score from input set

MLE (maximum likelihood estimator) $\hat{\theta} = \hat{\theta}(X)$. We can then find the MLE of $p(s)$, $\hat{p}(s) = \hat{p}(s, X)$, by using the popular plug-in method: $\hat{p}(s) = 1 - F_{\hat{\theta}}(s)$, where F_θ is the 3-Gamma CDF (cumulative distribution function).

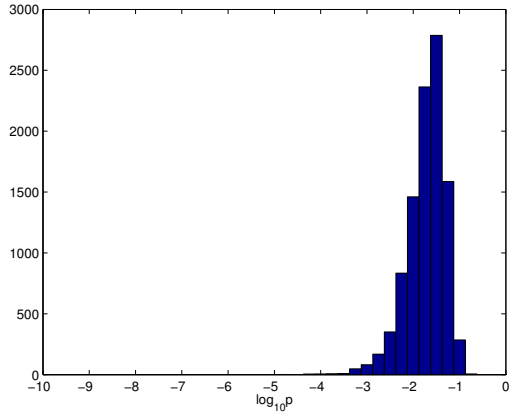
For a realistically small[b] sample size such as $n = 20$, $\hat{p}(s)$ can grossly over-estimate the significance of the observed score s (as noted in (Keich and Ng, 2007)). This type of MLE estimation, albeit using the normal approximation, is used in ((Liu et al., 1995; Narlikar et al., 2007; MacIsaac et al., 2006; Harbison et al., 2004)). We suspect that it further inflated the significance of the observed scores beyond that due to the selection of the normal approximation (see Figure 6.1 and Section 6.3.2 for evidence).

Our conservative "confidence p-value", $\hat{p_c}(s, X)$, presented in (Keich and Ng, 2007) corrects the tendency of the point estimator $\hat{p}(s)$ to over-estimate the 3-

[b] A sample of size n increases the runtime by a factor of n.



(a) $\hat{p}_n$overestimates the significance



(b) $\hat{p}_c(s, X)$is mostly conservative

Figure 6.1: **Comparing the estimators $\hat{p}_n$ and $\hat{p}_c(s, X)$ of p-value**$= 10^{-3}$. Histograms of 10^4 independent evaluations of the point estimator $\hat{p}_n(s)$ and of the conservative $\hat{p}_c(s, X)$, where s was set to the empirical 0.999 quantile. $\hat{p}_n$ is the MLE plug-in estimator of the p-value assuming a normal approximation, and $\hat{p}_c(s, X)$ is our conservative "confidence p-value" assuming a 3-Gamma distribution. The quantile s was learned from the scores of GibbsMarkov on 10,000 resampled sets of 30 sequences each of length 1,000. The resampling was done from the human genomic file. This set of null scores was then used to create the 10,000 resamples X of size $n = 20$ drawn with repetitions. An ideal estimator of $p(s)$ should have all the mass concentrated on the point -3 because s was set to the 0.999 quantile. It is clear from the graphs that $\hat{p}_n$ has a considerably larger variance than $\hat{p}_c$ and that it can badly over-estimate the significance of the score s. GibbsMarkov was run in OOPS mode with the parameters `-l 23 -gibbsamp -best_ent -t 170 -L 100 -em 0 -markov 3 -p 0.10`. Statistical estimations were done in R.

Gamma p-value, $p(s)$. It does so by constructing a confidence interval for the estimated $p(s)$. In principle, the confidence p-value can be applied whenever the 3-Gamma distribution is expected to offer a reasonably good fit to the finder's null distribution.

6.2 Factoring local base composition

Local base composition has long been taken into consideration in sequence analysis. For example, isochores are taken into account in the GENSCAN gene finding tool (Burge and Karlin, 1997). A considerable effort was made into incorporating sequence composition in pairwise local alignment significance analysis (e.g., (Altschul et al., 2005)). Another example is the motif finder NestedMICA incorporating a "mosaic background" model. The latter is a mixture of several, differently parametrized, low order, Markov chains which allow one to factor in local composition (Down and Hubbard, 2005). Regardless of whether or not our finder incorporates such mixture models, we argue here that the local composition should be taken into account when analyzing the significance of its results. Intuitively, imagine a set of sequences containing stretches made only from `A`. In this case a motif such as `AAAAAAAA` should not be too surprising.

We can factor local, or any other, composition information in our significance analysis in a rather straightforward manner. In principle, all we need to do is to condition our generated random sets on the relevant set of constraints. If the null distribution of the finder's score on these conditioned sets can be well approximated by the 3-Gamma distribution, then our confidence p-value method should be valid. Having no theory that could justify this approximation we resort to the

empirical studies as we previously did. Indeed, we can simply think of our conditional generating model described below as just another null set generator. Figure 6.2 compares the normal with the 3-Gamma approximation of such a conditional empirical null distribution.

Technically, our local GC-content adjusted resampling is done as follows (see Figure 6.3). We first divide our genomic reference file into partially overlapping windows of a fixed size L (overlap size is $L/2$). We then place each window in one of K bins that uniformly cover the entire spectrum of GC-content. This preprocessing step need only be done once. Given an input set we generate local GC-content adjusted resampled images of it as follows. We first divide each sequence into non-overlapping windows of size L and determine their GC-content. We then replace each of the original windows with a randomly drawn genomic window from the appropriate bin. Note that within a set we draw windows without replacement as repetitive elements can wreak havoc on motif finding. For the same reason we exclude overlapping windows within a set. The same kind of exclusion applies to our "uniform" resampling strategy.

Does factoring local GC content make a difference in the significance analysis? We give two different types of evidence that it does. First, Figure 6.4 compares histograms of our GibbsMarkov run on null sets that were generated according to the two models we are comparing. One model was generating sets using uniform resampling of a *S. cerevisiae* intergenic file while the other was using the local GC content framework described above. Notice that the two histograms are distinctly different. For example, a score whose p-value, when factoring in local GC content, is 0.0002 has a p-value of only 0.001 when assuming the uniform model.

As we just saw, taking into account the local GC-content can considerably

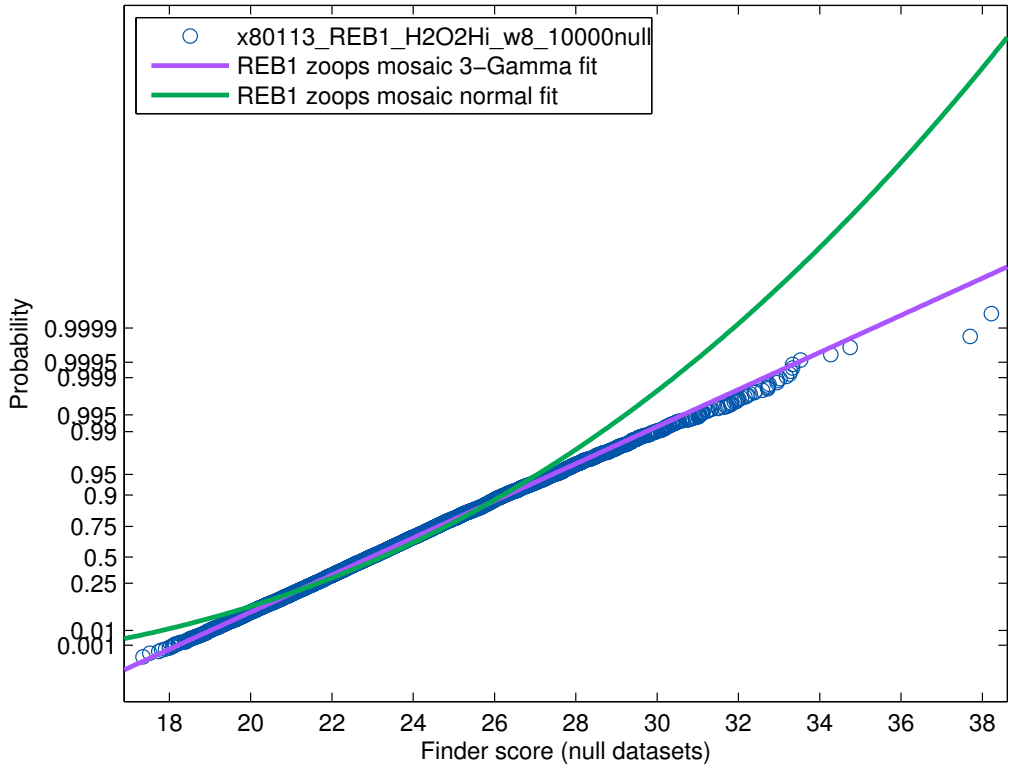


Figure 6.2: **Approximating a finder's null distribution conditioned on local GC-content.** The figure demonstrates the difference between the quality of the normal and the 3-Gamma approximations to a finder's null distribution. In this example, GibbsMarkov was applied to 10,000 sets of GC-content adjusted resampled sequences ($L = 100, K = 20$). The sequences were resampled from the *S. cerevisiae* intergenic file (see 6.4.3). The mold, or input, set was the Harbison REB1_H2O2Hi dataset consisting of 48 sequences of average length 431bp Harbison et al. (2004). The 3-Gamma seems to offer a reasonably good fit for this conditional null distribution while the normal does not. GibbsMarkov was run in ZOOPS mode with the parameters `-l 8 -gibbsamp -p 0.05 -best_ent -cput 300 -L 200 -em 0 -markov 5 -r 1 -ds -zoops 0.2`

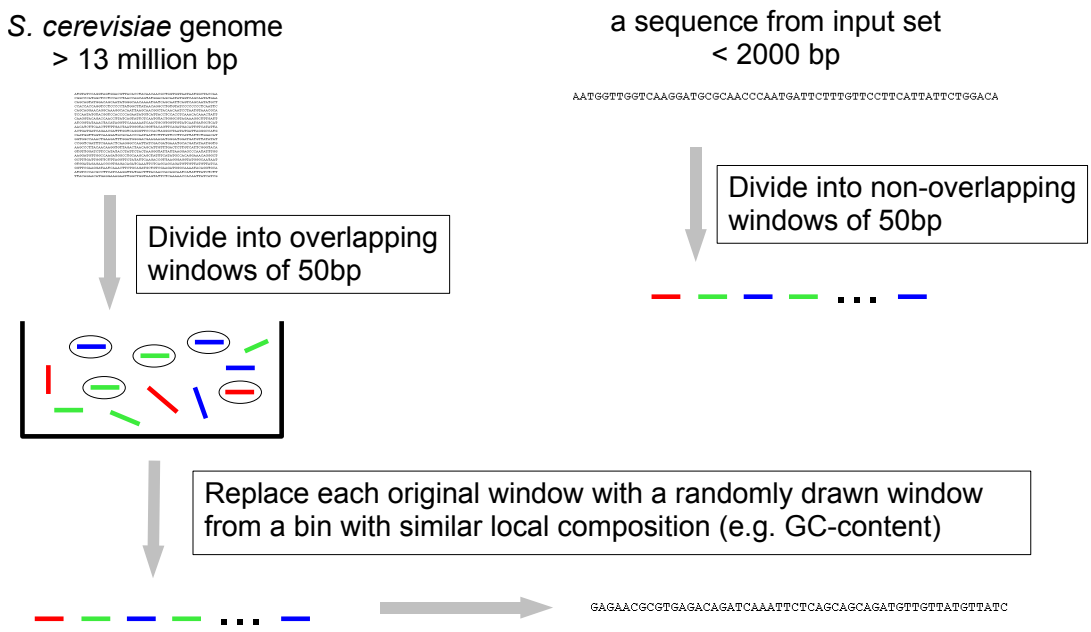


Figure 6.3: **Incorporating local GC content.** Suppose we have *S. cerevisiae* as our reference genome and we would like resample a set of local GC-content adjusted sequences based on a sequence from motif input data.

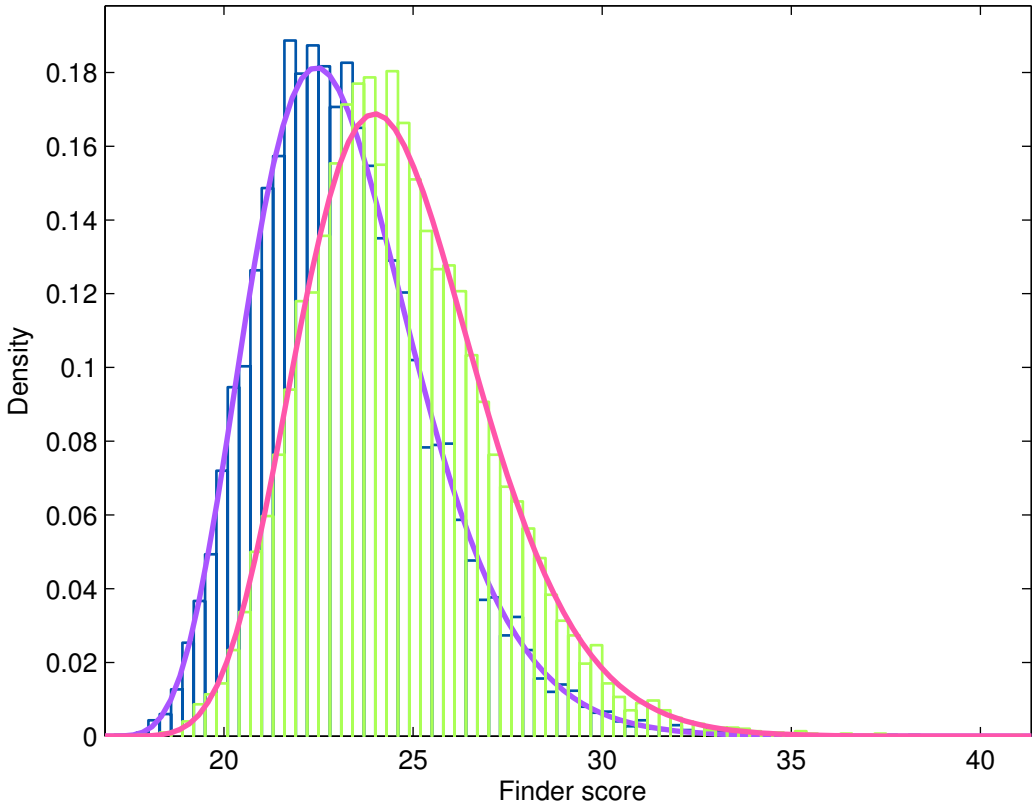


Figure 6.4: **Comparing the uniform and the local composition aware null generators.** The data for "right" histogram was generated by applying GibbsMarkov to 10,000 sets that were resampled uniformly from the *S. cerevisiae* intergenic file (see 6.4.3). The "left" histogram was generated using the same local GC-content preserving scheme as described for Figure 6.2. To highlight the difference both histogram were ML-fitted with a 3-Gamma distribution. GibbsMarkov was run in ZOOPS mode with the parameters `-l 8 -gibbsamp -p 0.05 -best_ent -cput 300 -L 200 -em 0 -markov 5 -r 1 -ds -zoops 0.2`

Figure 6.5: ACS motif

impact the significance of an observed score s. To demonstrate the potential difference between such a naive approach and our local GC-content adjusted one we devised the following experiments. This experiment is realistic in the sense that it emulates a real problem we encountered when analyzing DNA replication origins in *Saccharomyces kluyveri*. We first generated 200 random datasets by resampling from our human genomic file (see Section 6.4). To make these sequences look closer to the *S. kluyveri* sequences we were analyzing, we accepted only sequences whose AT-content is above 65%. We then implanted in each sequence exactly one site generated from the *Saccharomyces cerevisiae* AT-rich ACS profile[c] (see Figure 6.5). We next ran our GibbsMarkov in OOPS (one occurrence per sequence) mode on each of these 200 datasets, and noted the score, as well as whether or not the finder succeeded in uncovering the implanted ACS motif. Finally, we computed confidence p-values for each of these 200 scores in two different ways. The first was derived from our previous approach of uniform genomic resampling[d]. The second was derived from the new local GC-content preserving resampling scheme. Table 6.1 summarizes the results. Notably, the latter identifies 50% more TPs. The FPs

[c]The ACS is a 17bp site to which the *S. cerevisiae* ORC (origin recognition complex) binds to initiate local chromosomal replication (Sclafani and Holzen, 2007). We expect its *S. kluyveri* analogue to be somewhat similar.

[d]For technical reasons we used the same human genomic file which has roughly the same AT-level as that of *S. kluyveri*.

Table 6.1: **The effect of base composition on significance analysis.** The first number in each entry is the number of sets (out of 200) for which the p-value derived from sets generated by a uniform genomic resampling (57% AT-content). The second number is for the locally adjusted p-value. Notably, the latter identifies 50% more TPs. The overall high number of FNs is partly due to the conservative nature of the confidence p-value and partly due to the fact that these sets were designed as twilight zone ones.
Each of the 200 implanted sets consists of 30 sequences of length 2500 resampled from the human genomic file conditional on having an AT-content ≥ 65. Each sequence was implanted with exactly one site generated by drawing from the ACS matrix. This ACS matrix (Figure 6.5) was generated by us from a compiled list of confirmed ARSs on OriDB (Nieduszynski et al., 2007). GibbsMarkov was run in OOPS mode with the parameters `-l 17 -gibbsamp -p 0.05 -best ent -cput 300 -L 200 -em 0 -markov 3 -r 1`. The confidence p-values were derived from sets resampled in two different ways. Both resampled from our human genomic file but one conditioned the resampling on the local GC-content observed in the input dataset. Note that each one of these 200 input sets had a different local GC-content pattern.

p-value threshold	TP	TN	FP	FN
0.1	26/49	78/77	0/1	96/73
0.05	21/33	78/78	0/0	101/89

are under control in both cases as expected.

6.3 Results on Yeast ChIP-chip data

All the tests below refer to the Harbison dataset of 310 ChIP-chip, genome-wide location analysis, experiments of 203 yeast transcription factors (Harbison et al., 2004). By the "Narlikar test" we refer to the dataset consisting of the 156 sequence-sets from 80 TFs used in (Narlikar et al., 2007). The literature consensus for each of these 80 TFs is published. We obtained these from (Harbison et al., 2004), with the exception of DAL82, RTG1, and the modified CIN5 which we took from (MacIsaac et al., 2006). By the "MacIsaac test" we refer to the dataset consist-

ing of 188 sequence-sets which include all 124 TFs whose matrices are reported in (MacIsaac et al., 2006). See more details in Section 6.4. Unless otherwise noted, all significance analyses were performed using the local GC-content factoring technique that we described in Section 6.2

6.3.1 GibbsMarkov performance

We compared our motif finder GibbsMarkov with results from the Supplementary of (Narlikar et al., 2007). GibbsMarkov with fixed width $w = 8$ was run on the 156 sequence-sets. Using the same definition of success as defined in (Narlikar et al., 2007), GibbsMarkov successfully finds the correct motif in 71 of the 156 experiments. This is significantly better than all other *de novo* finders including PRIORITY-N (Narlikar et al., 2007) with 57 successes[e]. The full list which includes many more finders can be found in (Narlikar et al., 2007).

6.3.2 How well calibrated are these p-values?

If our p-values are well calibrated then the false discovery rate for any given threshold should be consistent with the rate guaranteed by the theory. To test that we applied the original FDR test (Benjamini and Hochberg, 1995) to find our p-value cutoff corresponding to an FDR of 5%. We applied this test separately to the p-values we assign to the 156 sets of the Narlikar test and then to the p-values we assign to the 188 sets of the MacIsaac test.

In order to get an accurate classification of predicted motifs, we disregarded

[e](Narlikar et al., 2007) reports that PRIORITY-N has 51 successes using a slightly different normalization. See Section 6.4.5.

motifs where (1) the consensus sequence of the predicted motif is AC-repeat or GT-repeat, and (2) the predicted motif does not match the literature motif but has a statistically significant match to a motif in the MacIsaac set of motifs (MacIsaac et al., 2006) (see Section 4.4 for details). Type (1) motifs which we found in ACE2 YPD, AFT2 H2O2Hi, ARR1 YPD, and SWI5 YPD were disregarded because GT-repeats are possibly functional in yeast ((Eden et al., 2007; Habib et al., 2008)). Type (2) motifs were disregarded because TFs often have co-factors that are DNA-binding. Such detected motifs should therefore not be considered false positives as they could still be biologically relevant.

At a 5% FDR threshold, for the MacIsaac test, our observed FDR comes at about 6.67%: 4/60, while for the Narlikar test it is about 7.41%: 4/54. At a 10% threshold, the observed FDR of the MacIsaac test and Narlikar test were 11.4% and 10.2%, respectively. Hence it is reasonable to conclude that our confidence p-values are well calibrated.

We also looked at the observed FDR of the results of (Narlikar et al., 2007) which are based on the normal MLE of the p-value. Their results were already disregarding the GT-repeats (type (1) from above), but we could not disregard possible type (2) motifs because we do not have access to their predicted motifs. At the 5% threshold their observed FDR on the Narlikar test is about 48%: 63/132, which is significantly higher than the expected 5%. For comparison, we repeated the FDR analysis on our confidence p-value by disregarding only the GT-repeats so that the comparison was on equal footing. At that 5% threshold, our observed FDR comes to about 12%: 7/57.

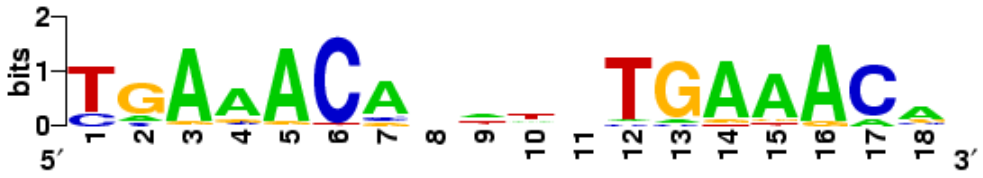
6.3.3 Ensemble: Using the p-value to improve our results

GibbsMarkov was run with multiple widths on the 156 sets of the Narlikar test, and a single predicted motif among the multiple widths was selected based on our confidence p-values. In the Narlikar test, our results improved from 71 successes with $w = 8$ to 76 with multiple widths. This is better than all other finders although PRIORITY-DN (Narlikar et al., 2007) which uses nucleosome positioning information is a close second with 75 successes[f]. The improvement was more significant in the MacIsaac test: the multiple widths method correctly identified 114 motifs while GibbsMarkov using $w = 8$ found only 97 out of 188 sequence-sets.

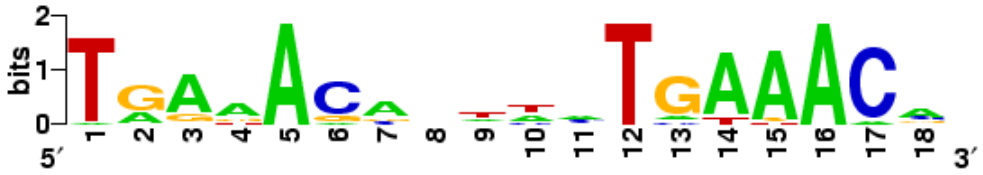
To test our performance of using confidence p-values for multiple widths selection, we compare it against naively selecting widths according to average entropy. Thus instead of choosing a predicted motif among widths with the best confidence p-value, a prediction is chosen based on average entropy, which is simply the entropy score averaged over the width of a motif. In the MacIsaac test, width selection based on average entropy found 99 while selection based on confidence p-values found 114 as reported above.

We have yet to thoroughly explore our predictions but one interesting dimer of width 18 caught our eyes. It appears essentially the same in three different experiments: DIG1 Alpha, TEC1 Alpha, and STE12 Alpha (see Figure 6.6). In all three cases width 18 exhibits the most significant p-value at: 3.7e-15, 1.3e-04, and 7.2e-08 respectively. A closer inspection shows the dimer is made of a repetition of the known motif common to DIG1 and STE12 (see Figure 6.7). This dimer was recently independently reported in (Habib et al., 2008).

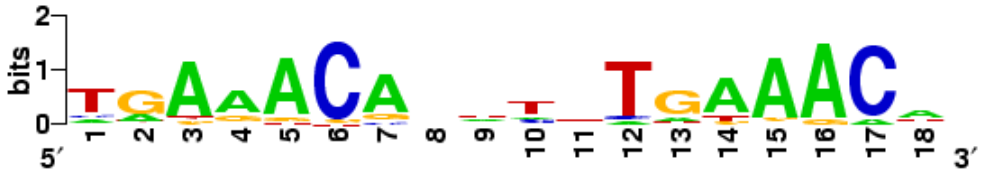
[f](Narlikar et al., 2007) reports that PRIORITY-DN has 70 successes using a slightly different normalization. See Section 6.4.



(a) DIG1 Alpha



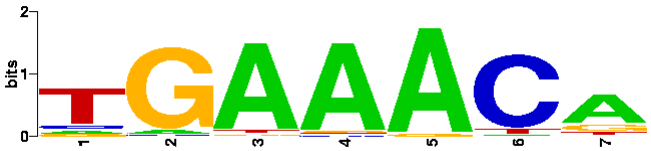
(b) TEC1 Alpha



(c) STE12 Alpha

Figure 6.6: Interesting dimer picked up GibbsMarkov

(a) DIG1



(b) STE12

Figure 6.7: Known motifs from (MacIsaac et al., 2006)

6.4 Methods of Yeast data experiments

6.4.1 Confidence p-values

All confidence p-values were computed in R (R Development Core Team, 2009) using functions described in (Keich and Ng, 2007). The necessary samples were derived from resampled data generated as described in the text.

6.4.2 GibbsMarkov

By GibbsMarkov we refer here to our variant of a Gibbs Sampler finder (Lawrence et al., 1993). Currently it handles an OOPS (one occurrence per sequence) or a ZOOPS (zero or one) model (Bailey and Elkan, 1995). A detailed account of

GibbsMarkov's sampling step and scoring function is described in Chapter 7.

6.4.3 Genomic files

We used two genomic files for resampling purposes. In both cases resampling was done by extracting contiguous sequences from a concatenated filtered genomic sequence. The "human genomic" contiguous sequence is from *Homo sapiens* chromosome 1 (HSA1). HSA1 was downloaded from the Ensembl Genome Browser v38 (NCBI build 36) (Birney et al., 2004). RepeatMasker, TandemRepeatFinder, and DUST were applied to the data. The *S. cerevisiae* intergenic file was generated by removing from the *S. cerevisiae* genome downloaded from SGD (Cherry et al., 1998) all protein and RNA coding sequences including tRNA, rRNA, snoRNA, snRNA, LTR, and other repetitive sequences.

6.4.4 Is the predicted motif a known motif?

Given a database of known motifs, we would like to determine whether a predicted motif has a statistically significant match to a known motif. For each predicted motif, we first obtained an empirical null distribution of maximal similarity scores (a higher score implies more similar motifs). Each score from this null is the maximal similarity score over all database PFMs against a random permutation of positions/columns of the predicted motif. Then the p-value for similarity is simply estimated from the null distribution described above and the similarity score between the predicted motif and its most similar motif within the database. Note that this technique accounts for evaluating statistical significance at the extreme value case of choosing the most similar motif within the database. In our FDR

analysis, the empirical null of each predicted motifs was generated with 10,000 randomly permuted motifs as described above and ignored cases where the predicted motif does not match the literature but has a p-value < 0.05 for similarity.

6.4.5 ChIP-chip dataset

All the consensus sequences were converted to PFM by the same method as (Harbison et al., 2004). For the MacIsaac tests, we used the same definition of success as defined in (Harbison et al., 2004). Likewise we used the definition of success defined in (Narlikar et al., 2007) for the Narlikar test with fixed width $w = 8$.

For the Narlikar test with *multiple widths*, we slightly modified the average entropy constraint of inter-motif distance used in (Narlikar et al., 2007). The average entropy of the predicted motif was taken over corresponding non-N positions of the literature consensus within an alignment, because predicted motifs such as GAL4 with literature consensus `CGGnnnnnnnnnnnCCG` should not be penalized for having degenerate positions at consensus positions with n.

GibbsMarkov was run with a fifth-order Markovian background estimated from the *S. cerevisiae* intergenic file. The strength of prior parameter in ZOOPS is $\alpha = 0.2$. The finder was allowed to run for 5 minutes with a plateau period of 200 iterations. All experiments were run under Red Hat Enterprise Linux 4 on a cluster with nodes that have AMD 248 2Ghz 64-bit processors with 2GB RAM and 1GB swap. The confidence p-values were computed from applying GibbsMarkov to 50 sequence-sets of local GC-content adjusted resampled sequences ($L = 100, K = 20$). For GibbsMarkov with multiple widths selection, GibbsMarkov parametrized with widths 8, 12, 15, and 18 were run separately on the input sequence-set, and

then each were applied separately on the same 50 sequence-sets of local GC-content adjusted resampled sequences.

Chapter 7

GIMSAN

We have demonstrated inherent flaws in the significance analysis based on the E-value of the information content (Chapter 5) as well as on the empirical normal approximation (Chapter 6). In contrast, we introduced a biologically realistic and reliable method to estimate the reported motif's statistical significance based on a novel 3-Gamma approximation scheme, as well as showed how we can further improve its reliability by factoring in local GC content (Chapter 6).

In this chapter, we present a novel *de novo* motif finding tool called GIMSAN (GIbbsMarkov with Significance ANalysis). GIMSAN combines GibbsMarkov, our variant of the Gibbs Sampler (Lawrence et al., 1993), with the aforementioned significance analysis. This tool is currently publicly available as a web application and a stand-alone application on Linux and PBS (Portable Batch System) cluster. A sequence logo of the detected motif is generated using the popular WebLogo (Crooks et al., 2004). In addition, GIMSAN tests whether the putative sites exhibit any pairwise positional dependencies.

GIMSAN allows the user to specify a range of motif widths. Recall in Section 6.3.3 we showed in similar such cases that selecting the optimal width based on our significance analysis can improve the results of de novo motif finding. Note, however, that choosing the best (i.e. lowest) P-value among several candidates amounts to multiple testing and a necessary correction should be employed by the user.

7.1 Significance evaluation

GIMSAN reports two figures that indicate the significance of the reported motif as outlined next. Based on the user selected reference set, GIMSAN generates null sets of sequences that preserve the dimensions and local GC content of the input set. It then runs GibbsMarkov with the user selected parameters on these null sets thereby creating a small sample of the finder's null distribution. Assuming this sample comes from a 3-Gamma distribution, GIMSAN reports a maximum likelihood point estimator of the p-value of the reported motif. Since the latter can significantly over-estimate the significance of the motif, GIMSAN augments it with a, roughly, 95% confidence interval of the p-value of the motif. For more details see (Keich and Ng, 2007) and Chapter 6.

7.2 Hybrid Gibbs sampler

By GibbsMarkov we refer here to our variant of a Gibbs Sampler finder (Lawrence et al., 1993). Currently it handles an OOPS (one occurrence per sequence) or a

ZOOPS (zero or one occurrence per sequence) model (Bailey and Elkan, 1995). Its scoring function and sampling steps follow the techniques presented in (Liu et al., 1995) and (Jensen et al., 2004). There are a couple of distinctions between these works and our implementation as described in Section 2.4. First, neither of the above papers specifically addresses the ZOOPS model described here. Second, these papers use a complete Bayesian framework which includes a prior on the matrices. Instead, we use a hybrid model which incorporates a prior on the percentage of sequences that include sites, but we use a maximum likelihood approach for the matrix. While the latter is fairly similar to using the Stirling approximation to the full Bayesian model (Jensen et al., 2004), it is not exactly the same. The ZOOPS model is specifically used in (Narlikar et al., 2007) but, again, there are some differences between the functions optimized there and ours. Specifically, our target function is different than theirs even in the case of uninformative prior they consider.

7.3 Motif column dependency

This section describes our technique to test dependency between any two motif columns from a given motif-finder's result. Its implementation is built into GIMSAN as a post-processing step to visualize motif positional dependencies (see Figure 7.1). Intuitively, column-pair dependency can be illustrated by the following

```
Hypergeometric p-values for statistically significant pairs (1 pairs)

Column-pair ( 6, 9) has estimated p-value 0.000000 [0.000000,0.000922] with ent=0.114 bits and nPerms=4000
O/E     A       C       G       T     ||        A          C          G          T    hypergeometric p-value
A      4/4     0/0     3/3     0/0    ||                                              A
C     46/36    0/0    19/29    0/0    ||   1.5e-004+             1.5e-004-           C
G      0/0     0/0     0/0     0/0    ||                                              G
T     12/22    0/0    29/19    0/0    ||   7.4e-005-             7.4e-005+           T

=============================================================================================================

Number of columns analyzed: 10 (out of 12)
Number of column-pairs: 45
Bonferroni corrected alpha level: 0.0011111

Column dependency p-values:
Column-pair ( 1, 2) has estimated p-value 0.742000 [0.713694,0.768873] with ent=0.007 bits and nPerms=1000
Column-pair ( 1, 4) has estimated p-value 0.233000 [0.207116,0.260466] with ent=0.020 bits and nPerms=1000
Column-pair ( 1, 5) has estimated p-value 0.347000 [0.317484,0.377420] with ent=0.038 bits and nPerms=1000
Column-pair ( 1, 6) has estimated p-value 0.595000 [0.563832,0.625607] with ent=0.019 bits and nPerms=1000
Column-pair ( 1, 7) has estimated p-value 1.000000 [0.996318,1.000000] with ent=0.009 bits and nPerms=1000
Column-pair ( 1, 9) has estimated p-value 0.805000 [0.779060,0.829130] with ent=0.004 bits and nPerms=1000
Column-pair ( 1,10) has estimated p-value 0.615000 [0.584038,0.645282] with ent=0.007 bits and nPerms=1000
Column-pair ( 1,11) has estimated p-value 0.376000 [0.345881,0.406851] with ent=0.027 bits and nPerms=1000
Column-pair ( 1,12) has estimated p-value 0.434000 [0.403014,0.465375] with ent=0.039 bits and nPerms=1000
Column-pair ( 2, 4) has estimated p-value 0.329000 [0.299918,0.359092] with ent=0.012 bits and nPerms=1000
Column-pair ( 2, 5) has estimated p-value 0.744000 [0.715757,0.770798] with ent=0.008 bits and nPerms=1000
```

Figure 7.1: GIMSAN column-dependency output

example of motif columns:

$$C^1 = \begin{bmatrix} \texttt{A} \\ \texttt{A} \\ \texttt{C} \\ \texttt{C} \end{bmatrix} \quad C^2 = \begin{bmatrix} \texttt{T} \\ \texttt{T} \\ \texttt{G} \\ \texttt{G} \end{bmatrix} \quad C^3 = \begin{bmatrix} \texttt{T} \\ \texttt{G} \\ \texttt{T} \\ \texttt{G} \end{bmatrix}$$

In this toy example, C^1 and C^2 have a higher pairwise dependency than C^1 and C^3 because a `T` appears in C^2 if and only if `A` appears in C^1 within the same w-mer. Specifically, we are given the input $\{C^i\}_{i=1}^{w}$ that represents a set of w motif columns of length N, where N is the number of motif sites. The statistical test for independence between each column-pair C^i and C^j is as follows. Denote I as our statistics that measures the degree of dependency between two columns:

$$I(C^i, C^j) := \sum_{a,b \in \{A,C,G,T\}} g^{i,j}_{a,b} \cdot \log \left(\frac{g^{i,j}_{a,b}}{f^i_a \cdot f^j_b} \right)$$

where $g^{i,j}$ is the joint probability frequency matrix of C^i and C^j, and f^i and

f^j is the probability frequency vector for C^i and C^j, respectively. We can then compute the p-value for testing independence using a simple resampling procedure. Technically, let $\{x_1, \ldots, x_n\}$ be a set of random permutations of column C_j and let $I_k := I(C^i, x_k)$. The point estimator can then be given as $\hat{p} := \frac{\sum_{k=1}^{n} \mathbf{1}_{I_k \geq I(C_i, C_j)}}{n}$, where $\mathbf{1}_{(\cdot)}$ is the indicator function.

In order to have a reasonable statistical power for multiple testing, we only test dependency on m non-degenerate columns, where non-degenerate columns are ones that have information content[a] between 0.25 and 1.75. Hence, for a given significance level α, the Bonferroni corrected level is $T := \alpha / \binom{m}{2}$. Let C be the 95% binomial confidence interval of the p-value, which can be computed from n and the point estimator $\hat{p}$ described above. If the upper limit of C is less than T, then our test rejects the null hypothesis of independence. If the lower limit of C is greater than T, then our test does not reject the null hypothesis. Otherwise, if T lies within C, then the test is repeated with twice as many random permutations. Intuitively, the doubling amount of the number of random permutations narrows the binomial confidence interval of the p-value to give a more reliable test.

7.4 User interface

GIMSAN is available as a Linux application and a web application (Figure 7.2). Description of the options for the web application interface shown in Figure 7.2 and their corresponding command-line options are as follows:

- **input FASTA** (–f): a set of sequences in FASTA format for de novo motif

[a] see Equation 1.2 and (Crooks et al., 2004) for details

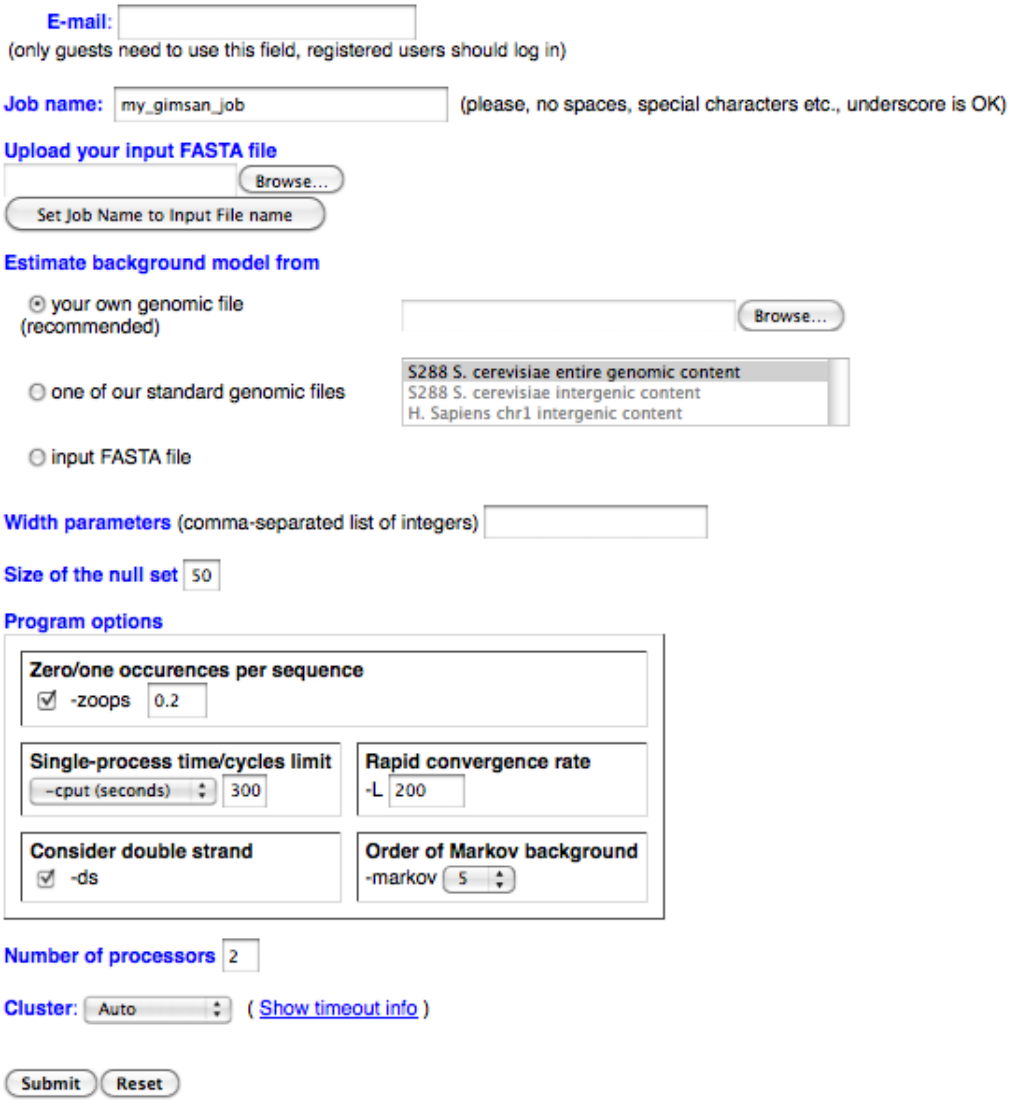


Figure 7.2: GIMSAN web interface

GIMSAN output from job **FHL1_YPD.fsa**

Run parameters:

Input file	FHL1_YPD.fsa
Size of the nullset	10
ZOOPS prior weight	0.2
Process run time limit	-cput 300
Convergence rate	200
Consider double-strand	yes
Order of Markov background	5
Genomic file	S288 S. cerevisiae intergenic content

---

span: 12, logo constructed from 113 sequences

The MLE of the p-value is 1.1e-73 and its confidence interval is (0,1e-43)

Column pairs with statistically significant dependency (1 pairs)

Motif finder detailed output

Figure 7.3: GIMSAN output

discovery

- **background model** (–bg): FASTA file for background model estimation. For example, this can be a set of S. Cerevisiae intergenic sequences. This data is used to generate null sets of sequences that preserve the dimensions and local GC-content of the input set, as well as estimating the background model for the de novo motif-finding task. Note: It is recommended that the user either "upload your own genomic file" or use "one of our standard genomic files".
- **motif widths** (–w): user can specify a range of motif widths (e.g. {8,14,20,30}). Once the GIMSAN job has completed, user can select the optimal width by choosing the motif with the lowest p-value (i.e. highest significance).
- **size of nullset** (–nullset): size of the randomly drawn set to estimate the motif-finder's null distribution based on 3-Gamma approximation. A larger null set would give a more accurate p-value at the expense of longer runtime.

Recall that the GIMSAN p-value is defined as the probability that a random sample of the same size as the input set will contain a motif of the same width that scores better than the motif found by GIMSAN. Thus a smaller p-value implies a more statistically significant motif. The MLE (maximum likelihood estimate) of the p-value in Figure 7.3 is 1.1×10^{-73}. Since this can often over-estimate the significance of the motif, GIMSAN augments it with a 95% confidence interval of the p-value. The upper value of the CI is the important number for discerning whether the candidate motif is statistically significant. In this example in Figure 7.3, the upper value of the CI is 10^{-43}. Note that the existence of duplicates and substring/superstring in the data could unduly inflate the significance of the reported motif.

Chapter 8

General framework for motif significance

In this chapter, we introduce MOTISAN (MOTIf finding with Significance ANalysis): a general framework for motif significance evaluation. Currently MOTISAN is implemented to perform analysis with a Gibbs sampling finder and the popular EM-based tool MEME (Bailey and Elkan, 1994). However, other motif finders can be easily incorporated into this flexible framework.

We will begin by showing that MEME's relative entropy score can indeed be well approximated by a 3-Gamma null distribution. Since MOTISAN allows the user to specify a range of finders and their parameters (e.g. motif widths), MOTISAN can be used as an ensemble algorithm that leverages our statistical significance scheme to select the optimal motif result. Furthermore, we will demonstrate in this chapter that MOTISAN p-values can be used as a width selection criteria to

improve the motif-finding task.

MOTISAN is publicly available as an application on Linux and PBS (Portable Batch System) cluster.

8.1 3-Gamma distribution fit for MEME

Previously, we have shown in Chapter 6 that the complete log-likelihood ratio of a Gibbs sampling finder can be well approximated by a 3-Gamma null distribution. Figure 8.1 shows that the 3-Gamma null distribution is a good approximation for MEME's relative entropy score as well.

MOTISAN reports two figures that indicate the significance of the reported motif as outlined next. Based on the user selected reference set, MOTISAN generates null sets of sequences that preserve the dimensions and local GC content of the input set. It then runs the user-specified motif finder on these null sets thereby creating a small sample of the finder's null distribution. Assuming this sample comes from a 3-Gamma distribution, MOTISAN reports a maximum likelihood point estimator of the p-value of the reported motif as well as a 95% confidence interval of the p-value of the motif. For more details see (Keich and Ng, 2007) and Chapter 6.

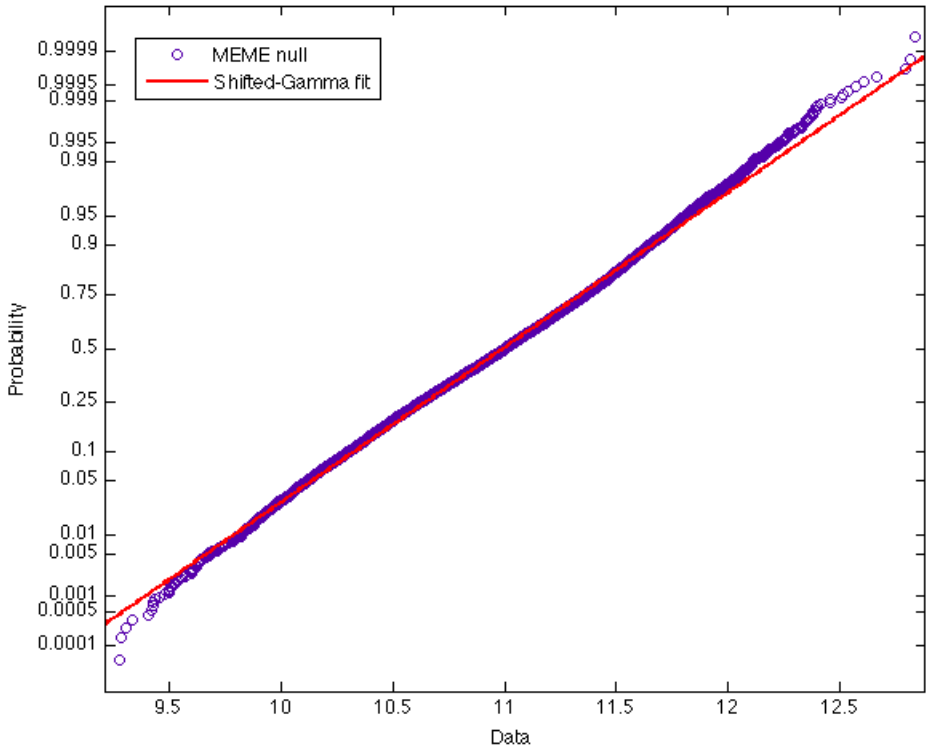


Figure 8.1: **Approximating MEME's null distribution.** The figure demonstrates the quality of the 3-Gamma approximations to a MEME null distribution. In this example, MEME was applied to 10,000 sets of GC-content adjusted resampled sequences. The sequences were resampled from the *S. cerevisiae* intergenic file. The mold, or input, set was the GAL4_YPD dataset consisting of 17 sequences of average length 506bp from Harbison et al. (2004). The 3-Gamma seems to offer a reasonably good fit for this conditional null distribution. MEME was run with the parameters: `-w 8 -dna -revcomp -text -mod oops -bfile yeast_Young_6k.3rd_order.markov`

8.2 Results from MOTISAN

8.2.1 How calibrated are MOTISAN's p-values?

The motif finders MEME and our Gibbs sampling finder GibbsMarkov (with fixed width $w = 8$) were run on the 156 sequence-sets that are defined in (Narlikar et al., 2007; Harbison et al., 2004) (see Methods). The number of successes is defined as the number of candidate motifs that have inter-motif distance[a] d less than 0.25 with literature confirmed motifs. MEME had 30 successes and GibbsMarkov had 64 successes. In addition, the finders were applied to 50 local GC-content preserving null sets for each of the 156 sets. To evaluate whether our estimated p-values are well calibrated, we examined the false-positive rates[b] by using "confidence p-values" (Keich and Ng, 2007) as the prediction metric (see Table 8.1). While the false positive rates are higher than they ought to be, keep in mind that a false positive here is not necessarily so in the pure statistical sense: some of the "negative" results are due to the finder's detection of a secondary *biologically significant* motif.

8.2.2 MEME's multiple width selection

We explore whether our MOTISAN p-values can be used as a width selection criteria to improve the motif-finding task. In particular, MEME was run on the 156 sets from (Narlikar et al., 2007; Harbison et al., 2004) across 8 different widths of $6 \leq w \leq 13$, and we examined the number of successes of all possible ensem-

[a] see Section 8.3 for details

[b] False-positive rate should be approximately equal to the p-value threshold if the p-values are well calibrated.

Table 8.1: MOTISAN's p-values and FP rates

The motif finders were run on 156 sequence-sets. To estimate p-values, the finders were applied to 50 local GC-content preserving null sets for each of the 156 sets. The false-positive rate (FPR), sensitivity (SEN), number of true-positives (TP), and area under ROC curve (AUC) are reported here.

	MEME (w=8) Motisan			GibbsMarkov (w=8) Motisan			MEME (w=8) E-value		
p-value threshold	FPR	SEN	TP	FPR	SEN	TP	FPR	SEN	TP
$p <$ 1E-5	0%	23%	7	2%	52%	33	-	-	-
$p < 0.01$	3%	57%	17	9%	78%	50	-	-	-
$p < 0.05$	9%	63%	19	13%	86%	55	-	-	-
AUC	0.869			0.915			0.853		
successes (out of 156)	30			64			30		

bles[c] consisting of non-singleton subsets of the 8 different widths. For each non-singleton subset of widths, we compared its MOTISAN's ensemble results with its best[d] performing individual finder (Table 8.2, Figure 8.2). Interestingly out of the 247 combinations, only 7 combinations (3%) are worse off when using MOTISAN's p-values as selection criteria under the stringent definition of *success* of $d < 0.15$ cutoff. Moreover, the ensemble MEME-MOTISAN has an improvement 95% (236/247) of the time compared over the best performing individual finder.

As for width selection criteria using MEME's built-in E-value, 60 combinations (24%) are worse off than the best performing individual finder, while 150 combinations (61%) are better off. Although the ensemble performance of MEME's built-in E-value and MOTISAN are comparable for $d < 0.25$ cutoff, the MOTISAN ensemble's advantage is pronounced under the more stringent $d < 0.15$ and $d < 0.10$ cutoff, as observed in Figure 8.2.

[c] a single predicted motif among the multiple widths was selected by choosing the one with smallest p-value

[d] choose the single width that has the highest number of successes on the 156 sets

Note that the aforementioned results were obtained by executing individual MEME runs across the 8 different widths rather than using MEME's minw/maxw option, which forces a continuous set of widths. Also note that the result with the least E-value among individual runs should be consistently better than using the minw/maxw (see Figure 8.3 for one such example) at the cost of running time.

Finally we remark that a typical user does not know the *best* individual width. Hence we compare MOTISAN's ensemble with the *median* performing individual finder (Figure 8.3), which is a more realistic scenario for a user that does not have *a priori* information about motif widths.

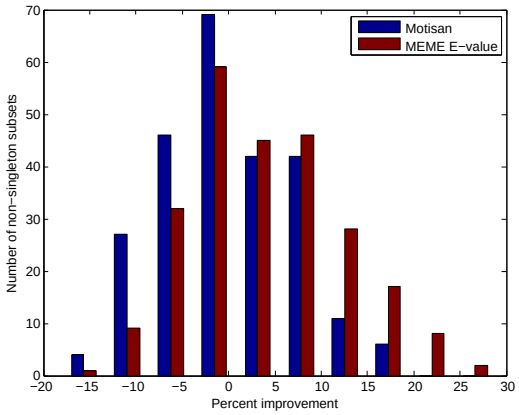
8.3 Methods for MOTISAN experiments

All the experiments that used GC content adjusted resampling scheme used L=50 and K=20, where L is the size of window and K is the number of bins (see Chapter 6 for details).

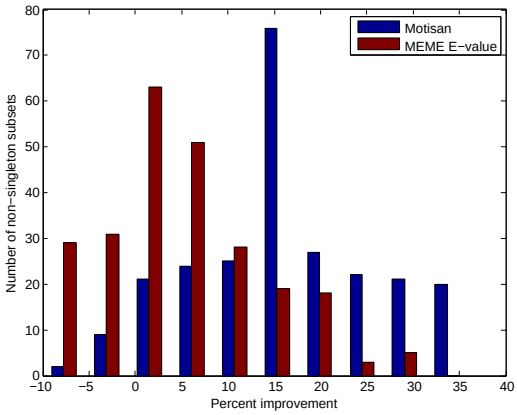
The yeast transcription factor binding data from ChIP-chip, genome-wide location analysis, experiments were obtained from (Harbison et al., 2004). The 156 motifs and their literature consensus in our motif-finding benchmark were obtained from (Gordân and Hartemink, 2008). For each of the 156 TF/condition sets, a motif input set composed of probes with binding p-value < 0.001.

Table 8.2: MEME ensemble. For a non-singleton subset of MEME widths, compare its ensemble results (either Motisan or MEME's E-value) with its best performing individual finder. There are 247 non-singleton subsets for 8 different widths of $6 \leq w \leq 13$. The "#(ensemble) − #(best)" column is defined as the number of successes (out of 156 sequence-sets) of a subset's ensemble minus its best performing individual finder. The table indicates the number of subsets that have the observed improvement with inter-motif distance d less than 0.1, 0.15 or 0.25.

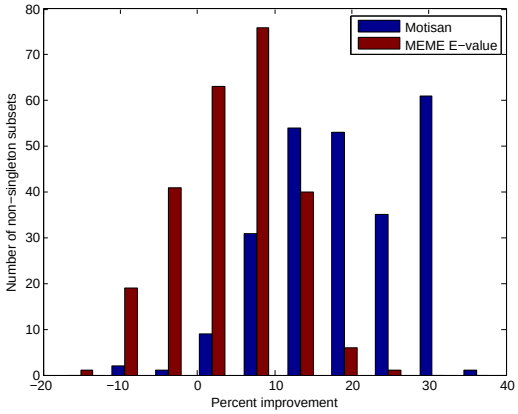
#(ensemble) − #(best)	**MEME-Motisan**			**MEME's E-value**		
	$d < 0.25$	$d < 0.15$	$d < 0.1$	$d < 0.25$	$d < 0.15$	$d < 0.1$
-7	1					
-6	2					
-5	7			2		
-4	8			6		
-3	19			14	10	1
-2	28	1	1	20	19	17
-1	27	6	5	32	31	43
0	43	4	7	27	37	63
1	38	15	26	31	48	76
2	35	20	50	33	33	40
3	24	42	99	27	28	6
4	9	52	51	20	21	1
5	5	55	7	17	15	
6	1	30	1	5	5	
7		16		7		
8		6		4		
9				1		
10				1		



(a) $d < 0.25$

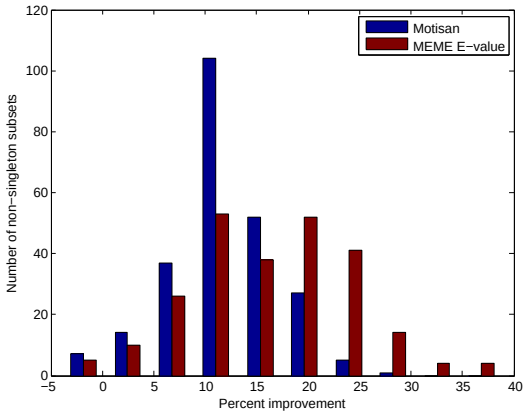


(b) $d < 0.15$

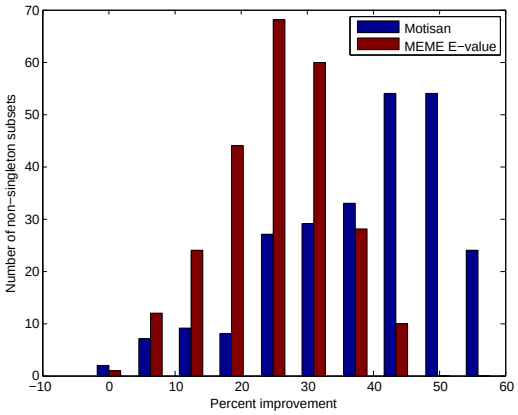


(c) $d < 0.1$

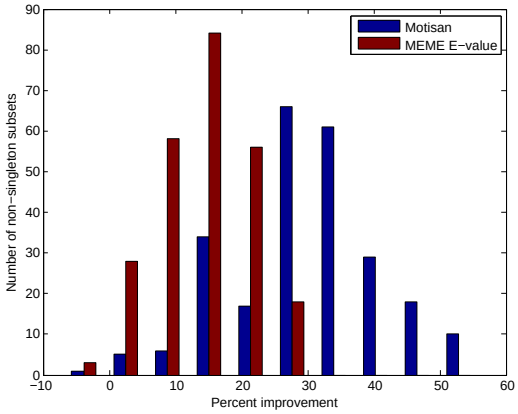
Figure 8.2: MEME ensemble compared with best individual component. For a non-singleton subset of MEME widths, compare its ensemble results (either Motisan or MEME's E-value) with its best performing individual finder. There are 247 non-singleton subsets for 8 different widths of $6 \leq w \leq 13$. The x-axis of the histogram shows the percentage of change in number of successes relative to the best performing individual finder, i.e. [#(ensemble) − #(best)] / #(best). The y-axis shows the number of non-singleton subsets for a particular bin. The number of successes is defined with respect to the inter-motif distance d between candidate motif and literature confirmed motif.



(a) $d < 0.25$



(b) $d < 0.15$



(c) $d < 0.1$

Figure 8.3: MEME ensemble compared with *median* performing individual component. For a non-singleton subset of MEME widths, compare its ensemble results (either Motisan or MEME's E-value) with its *median* performing individual finder. There are 247 non-singleton subsets for 8 different widths of $6 \leq w \leq 13$. The x-axis of the histogram shows the percentage of change in number of successes relative to the *median* performing individual finder, i.e. [#(ensemble) − #(median)] / #(median). The y-axis shows the number of non-singleton subsets for a particular bin. The number of successes is defined with respect to the inter-motif distance d between candidate motif and literature confirmed motif.

Table 8.3: MEME's performance. To determine the number of successes, we used the motif inter-motif distance d to compare the candidate with the literature confirmed motif (see Methods for details). The w column is the motif width parameter. The table shows the number of successes (out of 156 sets) for a given width parameter. **Note that MOTISAN's results are stochastic.

Finder	w	$d < 0.1$	$d < 0.15$	$d < 0.20$	$d < 0.25$
MEME	6	18	25	27	31
MEME	7	18	29	33	38
MEME	8	17	21	24	30
MEME	9	17	22	30	35
MEME	10	15	22	29	34
MEME	11	15	23	29	34
MEME	12	15	20	23	29
MEME	13	13	19	21	25
MEME least E-value	6-13	19	29	32	38
MEME minw/maxw	6-13	19	29	31	36
MEME-Motisan**	6-13	23	33	33	34

The inter-motif distance used in our benchmark is defined exactly as in (Gordân and Hartemink, 2008), which is a variant of the average root mean square error.

For MEME, all experiments in this chapter used the following parameters unless otherwise indicated: `"-dna -revcomp -mod oops"` with third-order Markov background model estimated from *S. cerevisiae* intergenic region. We used MEME's "relative entropy" as its score in our significance analysis. For GibbsMarkov, the parameters used were `"-cput 300 -L 200 -markov 5 -ds"`.

Chapter 9

Alignment constrained sampling

The development of tools for analysis of multiple sequences has played a central role since the advent of computational biology. Some examples include tools for multiple sequence alignment (e.g. (Thompson et al., 1994)), for motif-finding within a set of multiple sequences (e.g. (Wang, 2003)), and for construction of phylogenetic trees from multiple sequences (e.g. (Durbin et al., 1999)).

Given the complexity of these tools, statistical significance evaluations of their output is highly desirable. However, the very nature of multiple sequence analysis with its inter-dependency or correlated sequences makes any statistical analysis of these tools a daunting task.

In this chapter we suggest an approach to assigning significance to one class of problems involving correlated input of multiple sequences. Specifically we look at "homology-aware" motif finders such as MEME-C[a] (Bailey et al., 2010), Phylo-

[a]MEME with conservation position-specific prior (PSP). See Section 9.3 for details.

Con (Wang, 2003) and PRIORITY-C (Gordân and Hartemink, 2008) whose input consists of orthologous/homologous groups of sequences. These finders expect to find significant correlations or conservation within each homologous group, a fact which they take advantage of in order to guide their motif search.

Measures of statistical significance and in particular p-values are defined relative to a null model. The latter presumably captures the "non-interesting" parts of the data, or features that we expect the data to have regardless of whether or not it has some other distinction. A null model is required whether we compute our p-values using a theoretical model (e.g. (Altschul et al., 1990)) or derive them from an empirical sample (e.g. Chapter 6). In the latter case the null model is used to generate the non-interesting data points to which our analysis is applied (e.g. running a motif finder).

Ideally, a null model is an explicitly defined mathematical object however that might not always be possible to define. For example, how should the null model in the particular case we are interested in be defined? To begin with it is not clear what exactly is the case we are interested in other than the fact that the input sequences are not independently generated (which rules out standard naive random generators).

In this chapter, we present ALICO (ALIgnment COnstrainted) null set generator, which is a framework to generate randomized versions of the input alignment that preserve some of its crucial features including its dependence structure. An alternative empirical approach of "windowed alignment sampling" (WAS) was used to assign significance in (MacIsaac et al., 2006). Below we show that while WAS postulates a different null hypothesis than ALICO, at least in the case of the homology-aware finders we looked at, the two sampling methods yield very sim-

ilar results. Importantly, ALICO requires only *pairwise* alignment training data whereas WAS requires the full *multiple* alignment training data.

9.1 Generating random alignments that are "similar" to an input alignment

The basic component we would like to model is a block of a multiple alignment of d sequences. The input set can contain multiple such blocks in which case we assume these are independently generated. We chose not to assume that the alignment is generated by a phylogenetic tree and asked instead whether we can generate (pseudo) random alignments that would preserve some of the features of the input alignment. Specifically, we ask that our random alignments will have on average the same identity rates between its pairs of sequences as the PIDs (percent identities) between the corresponding sequences in the original input alignment.

In addition to these "vertical constraints" we are interested that each of our sequences on its own would resemble random genomic DNA. More specifically we would like its output to be "indistinguishable" from that of a k-th order Markov chain trained on some genomic training data[b]. We refer to these as the "horizontal constraints" and our goal is to generate random alignments that in some averaged sense satisfy this set of horizontal as well as vertical constraints.

It was not clear to us how to define an explicit probabilistic null model that would capture this set of goals. We therefore chose to approach this problem

[b]As discussed later, there are several plausible choices on which genomic data should this Markov chain be trained none of which is perfect.

in a somewhat ad hoc fashion, strengthening our constraints in one way while weakening it in another way so that we can define an inductive generative model that does satisfy the modified constraints. We then show how we can practically draw "approximate samples" from this model and argue based on the results below that our goals are largely met by this sampling method.

In the discussion that follows we *ignore the issue of gaps*, modeling instead only the gapless part of the alignment. The gaps are addressed though toward the very end of this discussion (Section 9.1.5.3).

9.1.1 A random pairwise alignment model

It is easy to define a model that would satisfy our original goals if we need to generate an "alignment" of one sequence. Indeed, a k-th order Markov chain estimated from our training data generates sequences that obviously satisfy our horizontal constraints.

The case of a pairwise input alignment is already exhibiting some of the difficulties in properly defining our problem. Assuming our input alignment comes from two species and assuming we have genomic training data for both, how shall we define our horizontal target chain: using the data of just one of the species, combining the data from both species, or maybe use a different horizontal target chain for each of the sampled sequences? In this chapter we gloss over this issue assuming the species variability in the horizontal chains is negligible.

Admitting this slack in our interpretation of the horizontal constraints our preliminary null pairwise alignment model is defined as a k-th order homogeneous Markov chain defined on aligned pairs of residues with transition probability func-

tion:

$$P\left(\begin{array}{c|ccc} X_i = x & X_{i-1} & \dots & X_{i-k} \\ Y_i = y & Y_{i-1} & \dots & Y_{i-k} \end{array}\right) = p_{2,k,k}\left(x, y | \boldsymbol{X}_{i-1:i-k}, \boldsymbol{Y}_{i-1:i-k}\right). \quad (9.1)$$

If the transition probabilities $p_{2,k,k}$ are estimated from a large pairwise alignment data then our pairs-emitting chain would generate a sampled alignment that arguably satisfies our horizontal constraints but not necessarily our vertical one: the average PID (percent identity) between the sampled sequences is determined by the training pairwise alignment. We therefore generate training data with the approximate target PID as follows.

Let $\rho = \rho(\boldsymbol{s}^1, \boldsymbol{s}^2)$ be our target PID: the PID between the two input sequences $\boldsymbol{s}^1$ and $\boldsymbol{s}^2$. We split the training pairwise alignment into windows and note the PID of each window. We then define n_b bins using the i/n_b quantiles of the observed PIDs ($i = 1, \dots, n_b$). Finally, we estimate the transition probabilities (9.1) from the bin whose range of PIDs includes ρ. Note that n_b can be adjusted according to the size of the training data where clearly the larger n_b is the smaller each bin's range is and therefore the closer is our sampled PID to the target ρ. That however needs to be balanced against the fact that we need to estimate (9.1).

While this solution for the pairwise alignment case approximately satisfies our vertical and horizontal constraints it is difficult to generalize it to deeper alignments. Learning a chain defined on d residues is quickly becoming infeasible for any k as the number of parameters[c] that need to be estimated is $4^{d(k+1)}$. For this reason we introduce the following alternative sampling procedure. The idea is to

[c]Aside from the fact that you need a training multiple alignment which we do not assume we have here.

alternately extend the two generated sequences by first sampling X_i followed by sampling Y_i according to the homogeneous conditional probabilities

$$\frac{P\left(X_i = x \,\middle|\, \begin{array}{ccc} X_{i-1} & \dots & X_{i-k} \\ Y_{i-1} & \dots & Y_{i-k} \end{array}\right)}{P\left(Y_i = y \,\middle|\, \begin{array}{cccc} X_i & X_{i-1} & \dots & X_{i-k} \\ & Y_{i-1} & \dots & Y_{i-k} \end{array}\right)}. \tag{9.2}$$

Note that for $d = 2$ the model described by (9.2) is identical to the one described by (9.1). In particular, trained on the same data[d], this procedure will generate essentially the same kind of samples for $d = 2$ as the chain on pairs defined in (9.1). However, (9.2) suggests how to proceed with the case of $d > 2$ sequences. Before we address the latter case we would like to introduce next yet another twist on our a pairwise alignment model.

The number of parameters we need to estimate in (9.2) is $\geq 4^{2k+2}$ which is typically infeasible for $k > 2$. Unfortunately, $k = 2$ is often not sufficient to capture the dependency observed in genomic DNA that badly affects the results of motif finders. At the same time our studies showed that given X_i and $Y_{i-1}, \dots, Y_{i-k}$ the dependence of Y_i on $X_{i-1}, \dots, X_{i-k}$ is rather weak, and the same goes for dependence of X_i on $Y_{i-1}, \dots, Y_{i-k}$ given $X_{i-1}, \dots, X_{i-k}$. Therefore we replace the conditional probabilities (9.2) governing our alternating sampling procedure with the following homogeneous conditional probabilities

$$\frac{P\left(X_i = x \,\middle|\, X_{i-1} \quad \dots \quad X_{i-k}\right)}{P\left(Y_i = y \,\middle|\, \begin{array}{cccc} X_i & & & \\ & Y_{i-1} & \dots & Y_{i-k} \end{array}\right)} = p_{1,1,k}\left(y \middle| X_i, \boldsymbol{Y}_{i-1:i-k}\right). \tag{9.3}$$

[d]These transition probabilities can be estimated from the same bin as the pairs chain of (9.1).

In this context when sampling Y_i we refer to sequence $\boldsymbol{X}$ as the "reference sequence" and note the implicit Markovian assumption: the present X_i or Y_i are independent of past data points given the values we condition on in (9.3).

The alternating model governed by (9.3) is not equivalent to the previous model described in (9.1) and again in (9.2). Specifically, while the vertical constraint is satisfied, up to bin accuracy, as above, the loosely defined horizontal constraints are now satisfied in a different manner. The advantage of this last model is that as the number of parameters that need to be estimated is of the order of 4^{k+2} we could readily go up to $k = 4$ in the examples described below.

9.1.2 A model for a random alignment of 3 sequences

Our goal is to define a model for a random alignment that, when properly parametrized, would satisfy our horizontal and vertical constraints. Importantly, we further restrict our attention to models that can be estimated using only pairwise alignment data. Therefore it is natural to consider inductive generative models: we define the conditional distribution of sequence $\boldsymbol{Z}$ given the random aligned pair $\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix}$. The joint distribution of the latter pairwise alignment can be defined through any of the null pairwise models of the previous section. In order to use higher order Markov chains in our sampling procedure, our pairwise alignment model in this chapter is the alternating model described in 9.3.

Possibly the simplest inductive model that comes to mind is to sample[e] $\boldsymbol{Z}$ using a single reference sequence, say $\boldsymbol{Y}$. Specifically, we sample Z_i using $p^{\boldsymbol{Y},\boldsymbol{Z}}_{1,1,k}\left(z\middle|Y_i, \boldsymbol{Z}_{i-1:i-k}\right)$

[e]In this section we found it convenient to define conditional probabilities in terms of sampling.

(9.3). The problem, of course, is that regardless of how we train $p_{1,1,k}^{\boldsymbol{Y},\boldsymbol{Z}}$ we have no direct control over $\rho(\boldsymbol{x},\boldsymbol{z})$, the PID between the sampled sequences $\boldsymbol{X}=\boldsymbol{x}$ and $\boldsymbol{Z}=\boldsymbol{z}$. In particular, $\rho(\boldsymbol{x},\boldsymbol{z})$ will generally not match $\rho(\boldsymbol{s}^1,\boldsymbol{s}^3)$, the observed PID between the two corresponding input sequences $\boldsymbol{s}^1$ and $\boldsymbol{s}^3$.

A more sophisticated model would randomly alternate between sampling using $\boldsymbol{X}$ and using $\boldsymbol{Y}$ as the reference sequence. That is, with a fixed probability $w \in [0,1]$ we sample Z_i using $p_{1,1,k}^{\boldsymbol{X},\boldsymbol{Z}}\left(z\middle|X_i,\boldsymbol{Z}_{i-1:i-k}\right)$ (9.3) and with probability $1-w$ we sample using $p_{1,1,k}^{\boldsymbol{Y},\boldsymbol{Z}}\left(z\middle|Y_i,\boldsymbol{Z}_{i-1:i-k}\right)$. Note that we are free to choose w as well as the similarity levels or the bins from which $p_{1,1,k}^{\boldsymbol{X},\boldsymbol{Z}}$ and $p_{1,1,k}^{\boldsymbol{Y},\boldsymbol{Z}}$ are estimated. Still, it is easy to construct examples that show this approach will typically fail as well. The issue is that the pattern of matches between the aligned residues is generally not independent. In particular, if $s_i^1=s_i^2$ then s_i^3 is typically more likely to match s_i^2 in real alignments than if $s_i^1 \neq s_i^2$. Accordingly, we devise a different model that tries to preserve the observed pattern of matches in the input set.

9.1.2.1 Model for preserving alignment columns partition frequencies

The pattern of matches of an alignment column is defined mathematically as the partition induced by the identity equivalence relation between the column residues. For example, there are two possible partitions for any pairwise alignment column: $\Pi^2=\{\{\{1,2\}\},\{\{1\},\{2\}\}\}$ where the partition is $\{\{1,2\}\}$ if $s_i^1=s_i^2$ and it is $\{\{1\},\{2\}\}$ if $s_i^1 \neq s_i^2$. Similarly, for a 3-letters column there are 5 possible partitions:

$$\Pi^3=\{\{\{1,2,3\}\},\{\{1,2\},\{3\}\},\{\{1\},\{2,3\}\},\{\{1,3\},\{2\}\},\{\{1\},\{2\},\{3\}\}\}$$

where the partition is $\{\{1,2,3\}\}$ if $s_i^1 = s_i^2 = s_i^3$, and it is $\{\{1\},\{2\},\{3\}\}$ if s_i^1, s_i^2, s_i^3 are all distinct, etc.

Our revised vertical constraint is to preserve the frequencies of the partitions of the input alignment columns. In general this is a much stronger condition than our original pairwise vertical constraint and, in particular, any model that satisfies our stronger constraint will satisfy our original goal. Note that for pairwise alignment preserving the frequency of the column partitions coincides with our previous constraint of preserving the pairwise PID. In particular, the pairwise methods described above are (approximately) preserving the partition frequencies. We therefore still consider an inductively defined model, effectively specifying how to sample $\boldsymbol{Z}$ given the aligned sampled $\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix}$.

For $m > j$ and partitions $\pi \in \Pi^j$ and $\sigma \in \Pi^m$ we say π is a sub-partition of σ, denoted $\pi \prec \sigma$, if the restriction of σ to $\{1,2,\ldots,j\}$ satisfies $\sigma\,|_{\{1,\ldots,j\}} = \pi$. Let Π_i denote the random partition (in Π^3) of column i that is generated by our (yet to be specified) model, and let $\Omega_i = \Pi_i\,|_{\{1,2\}}$ be its restriction to the first two residues of the column. It is not hard to see that our inductive model preserves the partition statistics if and only if

$$P\left(\Pi_i = \pi \mid \Omega_i\right) = 1_{\Omega_i \prec \pi} \cdot \frac{f_\pi}{f_{\Omega_i}} \tag{9.4}$$

where f_σ is the frequency of the partition σ in the input set. We next define a model that (approximately) satisfies (9.4). Our model is based on sampling Z_i given $\boldsymbol{Z}_{i-1:i-k}$ and either $\boldsymbol{X}_i$ or $\boldsymbol{Y}_i$ using (9.3) but we need to break it into cases as described next.

If $\Omega_i = \{\{1,2\}\}$ then we let Y_i be the reference sequence and sample Z_i using $p_{1,1,k}^{\boldsymbol{Y},\boldsymbol{Z},\Omega_i}\left(z \middle| Y_i, \boldsymbol{Z}_{i-1:i-k}\right)$. To satisfy (9.4) in this case $p_{1,1,k}^{\boldsymbol{Y},\boldsymbol{Z}}$ needs to be estimated

from pairwise alignment data with a target PID $c = f_{\{\{1,2,3\}\}}/f_{\{\{1,2\}\}}$. This can be approximately achieved by restricting attention to the bin whose range of PIDs include c.

If $\Omega_i = \{\{1\},\{2\}\}$ the sampling of Z_i is slightly more complicated: with probability w we choose $\boldsymbol{X}$ as the reference sequence and sample Z_i using $p_{1,1,k}^{\boldsymbol{X},\boldsymbol{Z},,\Omega_i}\left(z\middle|X_i, \boldsymbol{Z}_{i-1:i-k}\right)$ (9.3) whereas with probability $1-w$ we sample Z_i using $p_{1,1,k}^{\boldsymbol{Y},\boldsymbol{Z},,\Omega_i}\left(z\middle|Y_i, \boldsymbol{Z}_{i-1:i-k}\right)$. As explained next, the latter $p_{1,1,k}^{\boldsymbol{Y},\boldsymbol{Z},,\Omega_i}$ is in general estimated differently than the one we use for the case $\Omega_i = \{\{1,2\}\}$ above, specifically it typically has a different target PID.

Let ρ_{xz} and ρ_{yz} be the PIDs of the pairwise alignments from which the functions $p_{1,1,k}^{\boldsymbol{X},\boldsymbol{Z},\Omega_i}$ and $p_{1,1,k}^{\boldsymbol{Y},\boldsymbol{Z},,\Omega_i}$ are estimated. Then for $\pi = \{\{1,3\},\{2\}\}$ our model yields

$$P\left(\Pi_i = \pi \mid \Omega_i\right) \approx w\rho_{xz} + (1-w)\frac{1-\rho_{yz}}{3}. \tag{9.5}$$

The reason for the $\approx$ sign is that we made the simplifying assumption that conditioned on $\boldsymbol{Y}$ being the reference sequence and on $Z_i \neq Y_i$ (in addition to $X_i \neq Y_i$) the probability that $Z_i = X_i$ is approximately $1/3$ (it is exactly $1/3$ for uniformly distributed residues). Similarly, for $\sigma = \{\{1\},\{2,3\}\}$

$$P\left(\Pi_i = \sigma \mid \Omega_i\right) \approx (1-w)\rho_{yz} + w\frac{1-\rho_{xz}}{3}. \tag{9.6}$$

Comparing equations (9.5) and (9.6) with (9.4) we see that with $c_{xz} = f_\pi/f_{\{\{1,2\}\}}$ and $c_{yz} = f_\sigma/f_{\{\{1,2\}\}}$ the following system of equations are necessary and sufficient for our model to preserve the partition statistics:

$$\begin{aligned} w\rho_{xz} + (1-w)\frac{1-\rho_{yz}}{3} &= c_{xz} \\ (1-w)\rho_{yz} + w\frac{1-\rho_{xz}}{3} &= c_{yz} \end{aligned} \tag{9.7}$$

In addition to the freedom to choose $w \in [0, 1]$ we are also free to set the target PIDs[f] ρ_{xz} and ρ_{yz}. Therefore we have a system of 2 equations in 3 unknowns: the third equation coming from the partition $\{\{1\}, \{2\}, \{3\}\}$ is linearly dependent on the two equations of (9.7). To solve (9.7) we intuitively set $w = c_{xz}/(c_{xz} + c_{yz})$ and solve the resulting linear system for ρ_{xz} and ρ_{yz}.

A technical comment is in order here. While the determinant of the system is positive for $w \in (0, 1)$ the solutions ρ_{xz} and ρ_{yz} are not guaranteed to be positive. However, we established numerically that for "reasonable" alignment partition statistics where $\max\{c_{xz}, c_{yz}\} \geq 1/4$ and $\min\{c_{xz}, c_{yz}\} \geq (1 - \max\{c_{xz}, c_{yz}\})/3$ [g] the target PIDs ρ_{xz} and ρ_{yz} are in $[0.25, 1]$.

It is important to note that while our model (approximately) guarantees preserving the partition frequencies (vertical constraints) it does not explicitly address the horizontal constraints. It is however implicitly addressing them in a couple of ways:

- The first sequence $\boldsymbol{X}$ is sampled using only horizontal information (the Markov chain).
- For other sampled sequences the newly sampled residue is conditioned also on the last k residues of the sequence.

Our tests below show that this procedure generates alignments that are reasonably satisfying the horizontal (and vertical) constraints[h].

[f] Although in practice we can only approximate a desired PID due to the binning procedure.

[g] This condition simply states that a match in aligned positions is more likely than a random match.

[h] Obviously this is a subjective statement but the readers are invited to judge for themselves by inspecting the results of Section 9.2.1.

9.1.3 Sampling a random alignment of 4 sequences

Our model for generating random alignments of 4 sequences with given partition statistics is again defined inductively. We assume that, using the model described in the previous section, we can sample 3 sequences $\boldsymbol{X}^1, \boldsymbol{X}^2, \boldsymbol{X}^3$ with the required sub partition statistics. We next show how we can sample $\boldsymbol{X}^4$ given $\boldsymbol{X}^1, \boldsymbol{X}^2, \boldsymbol{X}^3$ so that the combined model preserves on average the given partition frequencies.

Let Π_i be the random partition (in Π^4) of column i that is generated by our model and let $\Omega_i = \Pi_i \mid_{\{1,2,3\}}$ be its restriction to the first three residues. For a partition σ let $|\sigma| \in \{1,2,3,4\}$ denote its number of classes, e.g., $|\{\{1,2,3\}\}| = 1$ and $|\{\{1\},\{2\},\{3\}\}| = 3$. For Ω_i with $|\Omega_i| = 1$ or $|\Omega_i| = 2$ the previous section on sampling 3 sequences already showed us how to sample $\boldsymbol{X}^4$ so as to preserve the input alignment partitions frequencies (see also Section 9.1.5 below). The case that remains is $|\Omega_i| = 3$ which given that $\Omega_i \in \Pi^3$ implies $\Omega_i = \{\{1\},\{2\},\{3\}\}$.

If $\Omega_i = \{\{1\},\{2\},\{3\}\}$ we sample X_i^4 as follows: with probability w_j ($j = 1,2,3$ and $\sum_1^3 w_j = 1$) we choose $\boldsymbol{X}^j$ as the reference sequence and sample X_i^4 using $p_{1,1,k}^{j,4,\Omega_i}\left(x \middle| X_i^j, \boldsymbol{X}_{i-1:i-k}^4\right)$ (9.3). Denoting by ρ_j the PID of the pairwise alignment from which the function $p_{1,1,k}^{j,4,\Omega_i}$ is estimated we have analogously to (9.5) and (9.6):

$$P\left(\Pi_i = \pi_j \mid \Omega_i\right) \approx w_j \rho_j + \sum_{l \neq j} w_l \frac{1-\rho_l}{3}, \tag{9.8}$$

where $\pi_1 = \{\{1,4\},\{2\},\{3\}\}$, $\pi_2 = \{\{1\},\{2,4\},\{3\}\}$, $\pi_3 = \{\{1\},\{2\},\{3,4\}\}$.

Note that the necessary and sufficient condition (9.4) for preserving the partition statistic holds as is in our case as well. Thus, our model preserves the partition frequencies iff the RHS of (9.8) matches the RHS of (9.4) which yields a system

of 4 equations in 6 unknowns:

$$w_j\rho_j + \sum_{l\neq j} w_l \frac{1-\rho_l}{3} = c_j \qquad j = 1,2,3 \tag{9.9}$$

where $c_j = f_{\pi_j}/f_{\Omega_i}$ (in addition to $\sum_1^3 w_j = 1$).

Intuitively setting $w_j = c_j/(c_1+c_2+c_3)$, the determinant of the resulting linear system (9.9) is positive if $c_j \neq 0$ for all j offering a unique solution for the target PIDs ρ_j. A similar remark for the case $d = 3$ sequences holds here: if the values of c_j are "reasonable" in the sense that matches in aligned positions are at least as likely as random matches[i] then again the solved target ρ_i are guaranteed to be in $[0.25, 1]$.

9.1.4 Sampling a random alignment of 5 sequences

Using the same framework as in the last couple of sections we only need to describe how to sample X_i^5 in the case where $\Omega_i = \{\{1\},\{2\},\{3\},\{4\}\}$: with probability w_j ($j = 1,2,3,4$ and $\sum_1^4 w_j = 1$) we choose $\boldsymbol{X}^j$ as the reference sequence and sample X_i^5 using $p_{1,1,k}^{j,5,\Omega_i}\left(x\middle|X_i^j, \boldsymbol{X}_{i-1:i-k}^5\right)$ (9.3). Using the same arguments as above we arrive at a system of 5 equations in 8 unknowns. Setting intuitively $w_j = c_j$ where $c_j = f_{\pi_j}/f_{\Omega_i}$ and $\pi_1 = \{\{1,5\},\{2\},\{3\},\{4\}\}$, $\pi_2 = \{\{1\},\{2,5\},\{3\},\{4\}\}$ etc., yields a linear system of 4 equations in 4 unknowns (ρ_i):

$$w_j\rho_j + \sum_{l\neq j} w_l \frac{1-\rho_l}{3} = c_j \quad \Longleftrightarrow \quad w_j\rho_j - \sum_{l\neq j} \frac{w_l}{3}\rho_l = c_j - \sum_{l\neq j}\frac{w_l}{3}$$

for $j = 1,\ldots,4$. By summing the LHS for all j we get 0 but the same thing holds for the RHS. This suggests there is a 1-dimensional affine subspace of solutions. Indeed

[i]The technical condition is $c_{(1)} \geq 1/4$, $c_{(2)} \geq (1-c_1)/3$, and $c_{(3)} \geq (1-c_1-c_2)/2$ where $c_{(i)}$ are the decreasingly ordered c_i.

the vector $(1,1,1,1)$ is a solution and it is not hard to see that if $w_0 := \min_j w_j = 0$ then it is the only solution. Otherwise, the general solution is given by

$$(1,1,1,1) + \alpha \cdot \left(\frac{1}{w_1}, \frac{1}{w_2}, \frac{1}{w_3}, \frac{1}{w_4}\right) \qquad \alpha \in [-w_0, 0].$$

In general we choose the smallest possible[j] α or, $\alpha = -w_0$. The reason for that is that $\rho_j \approx 1$ essentially means we ignore the horizontal information when using sequence $\boldsymbol{X}^j$ as the reference sequence.

9.1.5 ALICO (ALIgnment COnstrainted) sampling - the general case

All the elements of our model that have been introduced in the previous sections are now combined to show how we sample a random alignment of m sequences. The sampling is performed inductively or sequentially according to a predetermined sampling order (see below). We assume the sequences of the input alignment are ordered according to our sampling order and proceed as follows:

- The first sequence $\boldsymbol{X}^1$ is sampled according to a kth order Markov chain.
- Given the sampled sub-alignment $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^{m-1}$ we sample sequence $\boldsymbol{X}^m$ one position at a time. Our sampling procedure of X_i^m depends on the partition Ω_i defined by $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^{m-1}$. While each partition Ω_i defines its own sampling procedure, there are only 4 *types* of partitions depending on the value of $|\Omega_i|$:

[j]Technically it might be tricky to get sufficient pairwise alignment training data for that target PID. Therefore in practice we simply use $\alpha_0 \approx -3/4w_0$.

$|\Omega_i| = 1$) Use any of the $\boldsymbol{X}^i$ ($i \leq m-1$) as the reference sequence (they are all the same) and set the target PID to $f_{\{\{1,\ldots,m\}\}}/f_{\{\{1,\ldots,m-1\}\}}$.

$|\Omega_i| = 2$) Let $\boldsymbol{X} \in A$ and $\boldsymbol{Y} \in B$ be two sequences chosen from the two different classes of the partition $\Omega_i = \{A, B\}$, and let $c_{xz} = f_{\{A\cup\{m\},B\}}/f_{\{A,B\}}$ and $c_{yz} = f_{\{A,B\cup\{m\}\}}/f_{\{A,B\}}$. With probability w use $\boldsymbol{X}$ as the reference sequence and with probability $1-w$ use $\boldsymbol{Y}$. The target PIDs as well as w should be set according to the solution of (9.7) as described in that section.

$|\Omega_i| = 3$) Sample as described in Section 9.1.3 with $\boldsymbol{X}^j$ replaced by $\boldsymbol{X}^{n_j} \in A_j$ where $\Omega_i = \{A_1, A_2, A_3\}$ and $c_1 = f_{\{A_1\cup\{m\},A_2,A_3\}}/f_{\Omega_i}$, $c_2 = f_{\{A_1,A_2\cup\{m\},A_3\}}/f_{\Omega_i}$, and $c_3 = f_{\{A_1,A_2,A_3\cup\{m\}\}}/f_{\Omega_i}$.

$|\Omega_i| = 4$) Sample as described in Section 9.1.4 with $\boldsymbol{X}^j$ replaced by $\boldsymbol{X}^{n_j} \in A_j$ where $\Omega_i = \{A_1, A_2, A_3, A_4\}$ and
$c_1 = f_{\{A_1\cup\{m\},A_2,A_3,A_4\}}/f_{\Omega_i}$, $c_2 = f_{\{A_1,A_2\cup\{m\},A_3,A_4\}}/f_{\Omega_i}$ etc.

9.1.5.1 Sampling order

The order in which the sequences are sampled is determined as follows.

1. Create a list S and initialize it as an empty list. S represents sequences that have been chosen.
2. Choose the first two sequences to sample by selecting the pair with the highest PID and arbitrarily assign them the first two places in the list S.
3. Choose the next sequence to add to S as the sequence with highest average PID with respect to the sequences that are currently in S.
4. Repeat the last step until S contains all the sequences.

9.1.5.2 What to do about the huge number of partitions

The number of possible partitions for a column of d letters grows exponentially with d as it is given by (breaking it into the number of partitions per the number of classes of the partition):

$$1 + \frac{2^d - 2}{2} + \frac{3^d - 3 \cdot 2^d + 3}{3!} + \frac{4^d - 4 \cdot 3^d + 6 \cdot 2^d - 4}{4!}.$$

As long as d is fairly small, say $d \leq 5$, and the input alignment is of typical size (hundreds of columns) the method should work reasonably well exactly as described above. However, if $d \geq 6$ and the input alignment is not "unnaturally long" we might see many partitions appearing only once or twice. While in principle our method should be able to handle these cases correctly it might introduce some degeneracy into our sampling.

A possible solution to this problem that we have yet to explore is to relax the vertical goal of preserving the average frequencies of *all* partitions. In particular, instead of trying to preserve the frequencies of partitions that appear once or twice we bundle these together according to shared sub-partitions and try to preserve the frequencies of the bundled sub-partitions.

9.1.5.3 Handling gaps

Our approach to handling gaps is to leave them as they are. More specifically when we sample X_i^m we leave it as a gap if the original input alignment has a gap in this position. Otherwise, we sample as described above except that we ignore all the sequences that have a gap in column i.

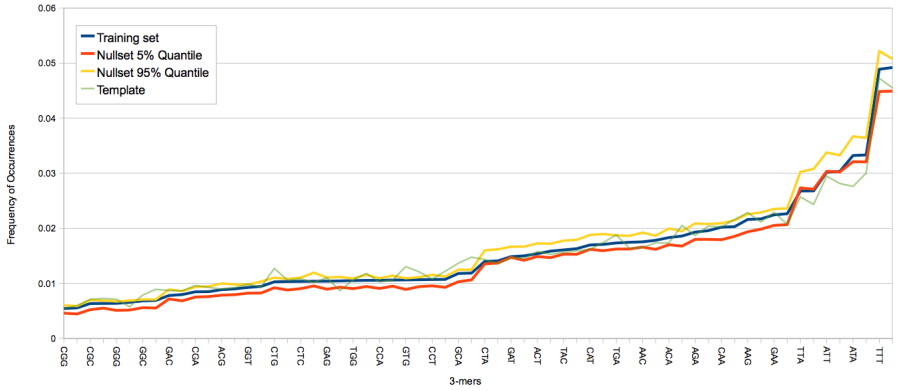
9.2 Results of Alignment Constrained Sampling

9.2.1 Satisfying the horizontal and vertical constraints

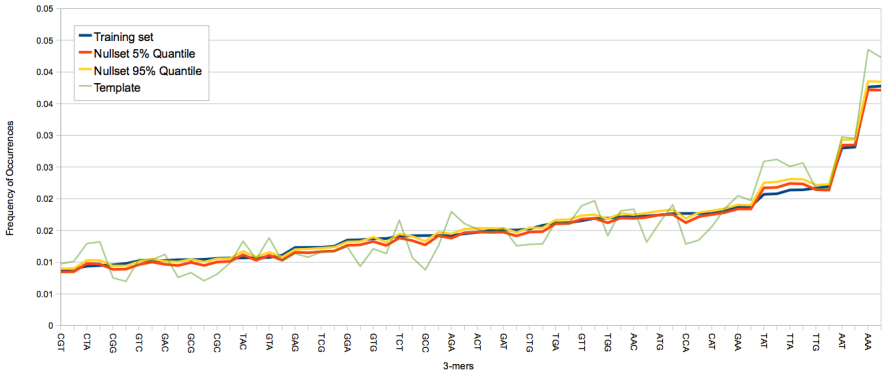
In this section we give examples showing that ALICO sampling gives null sets that essentially satisfy the horizontal as well as the vertical constraints. We generated 100 ALICO null samples with CAD1_YPD as the template (input set). The CAD1_YPD data has 20 orthologous groups, each with orthologous sequences from 4 different species, and has a total concatenated alignment length of 13276 (see Section 9.3). The pairwise PIDs of the template were compared with the ALICO sampled null sets as well as with "windowed alignment sampling" (WAS) which relies on randomly sampling windows from a large multiple alignment training data (see Figure 9.2). To show that the null sets satisfy the horizontal constraints, a 2nd-order Markov model in this case, we compared the frequency of 3-mers that appeared in the template and the null sets, as well as the training data (see Figure 9.1a).

9.2.2 Motif finders' parametric score distribution

We examine the distribution of the scores of PhyloCon, PRIORITY-C and MEME-C under the ALICO and WAS null model. Recall from Chapter 6 that if a finder's empirical null distribution can be well approximated by some parametric distribution, then we can reliably estimate p-values by simply estimating the parameters of its parametric distribution. Thus PhyloCon, MEME-C and PRIORITY-C were applied to 10,000 sets of ALICO sampled and WAS sampled alignment sets with GAL4_YPD as template (see Figure 9.3, 9.5, 9.6). Note that the normal distri-



(a) Horizontal constraint of 100 null sets using CAD1_YPD (yeast) as template



(b) Horizontal constraint of 100 null sets from Drosophila (6 species)

Figure 9.1: Horizontal constraint of 100 null sets. The frequencies of 3-mers are compared between the training data, template (input set), and 100 ALICO sampled null sets. The 3-mers on the x-axis are sorted according to their frequency of occurrences within the training data. For the 100 ALICO-generated samples, we computed the 5%-quantile and 95%-quantile of occurrence frequencies for each of the 3-mer. (a) CAD1_YPD from yeast as template. (b) template input alignment of six Drosophila species (see Section 9.3).

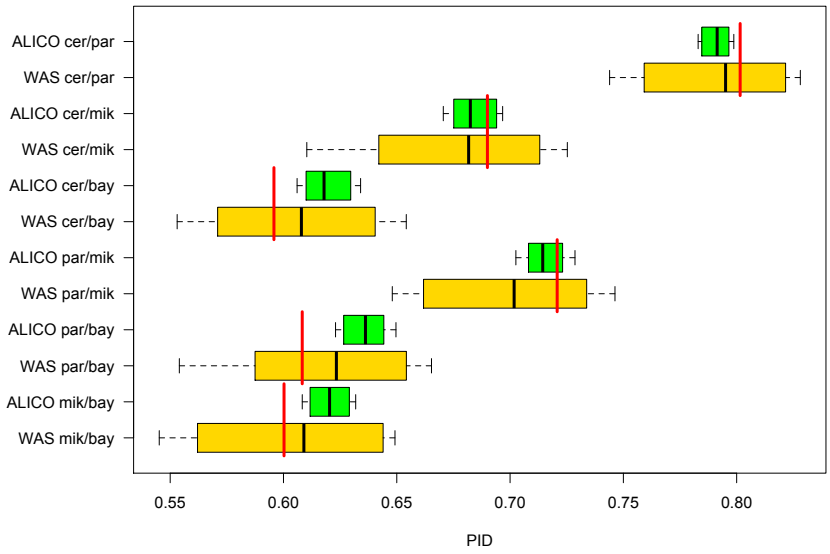


Figure 9.2: Vertical constraint satisfaction in 100 null sets using CAD1_YPD (yeast) as template (input set). The red vertical lines indicate the observed pairwise PIDs of the template. Each box and its whiskers show the minimum, 5%-quantile, median, 95%-quantile, and maximum PIDs of the sampled null sets. The ALICO samples occasionally exhibit slightly increased bias when compared with the pairwise PIDs observed in the WAS null sets. At the same time the variance of these PIDs is significantly reduced in the ALICO samples.

bution and 3-Gamma seem to offer a reasonably good fit for PhyloCon under our ALICO null model, but we were not able to find a parametric fit for PhyloCon under the WAS model (see Figure 9.3b). The 3-Gamma offers a good fit for MEME-C and PRIORITY-C under both ALICO and WAS null model.

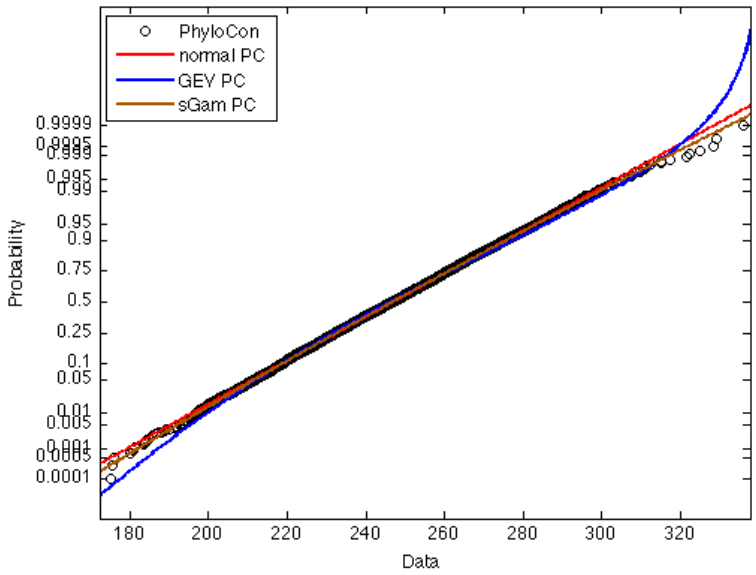
9.2.3 Comparison with WAS

ALICO postulates a different null model from the WAS null model. It is therefore interesting to compare the two in terms of the performance of the homology-aware motif finders[k]: PRIORITY-C and MEME-C. The finders were first applied to 10,000 sets of WAS sampled alignment sets with GAL4_YPD as template. Then the finders were applied to 10,000 ALICO null sets that were generated by sampling *one* ALICO set with each of the 10,000 WAS null in the previous step as template. We generated ALICO null sets using this method because ALICO preserves gap locations while WAS sampled nulls have different number of non-gap positions than the template. Therefore in order to compare apples to apples, the ALICO null sets were generated from WAS null sets so that both sets have equal dimension and number of non-gap positions. If the ALICO sampling resembles WAS, then the two score distributions for a particular finder should be similar (see Figure 9.4).

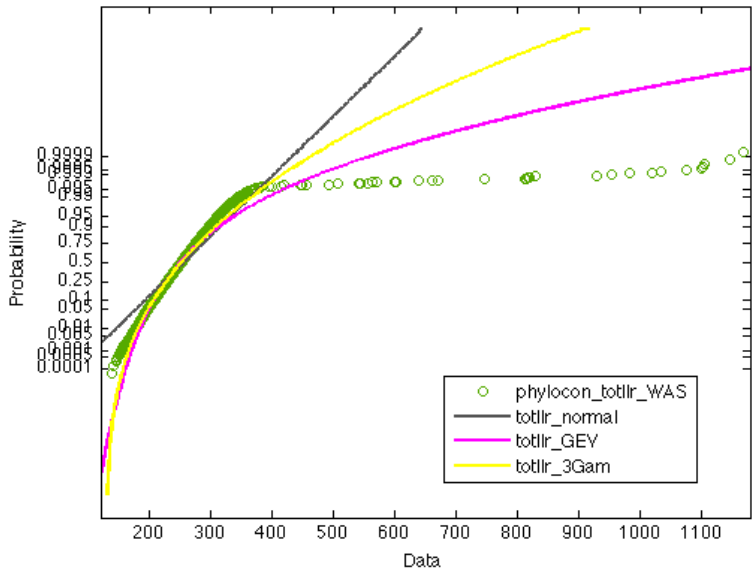
9.2.4 Are the ALICO p-values well calibrated?

The homology-aware motif finders PhyloCon, MEME-C, and PRIORITY-C (with fixed width w=8) were run on the 156 orthologous sequence-set that are defined

[k]PhyloCon was not considered in this comparison because its distribution is erratic under WAS model (see Figure 9.3b)

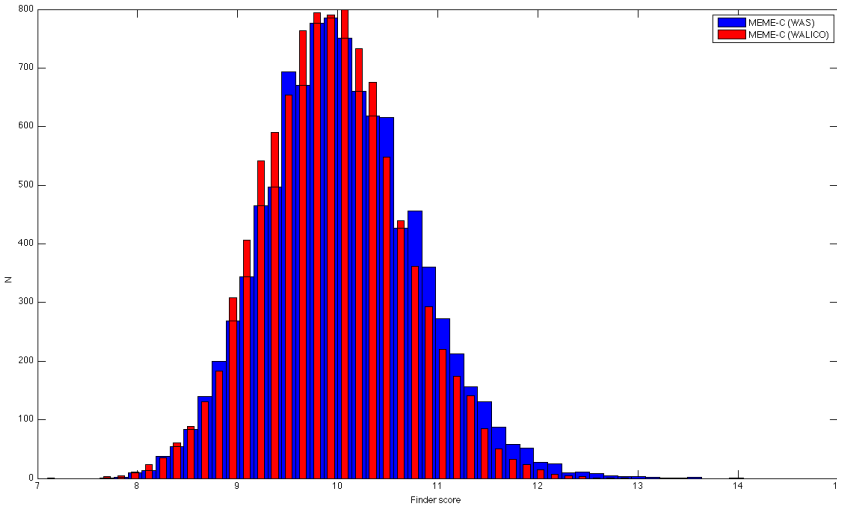


(a) ALICO

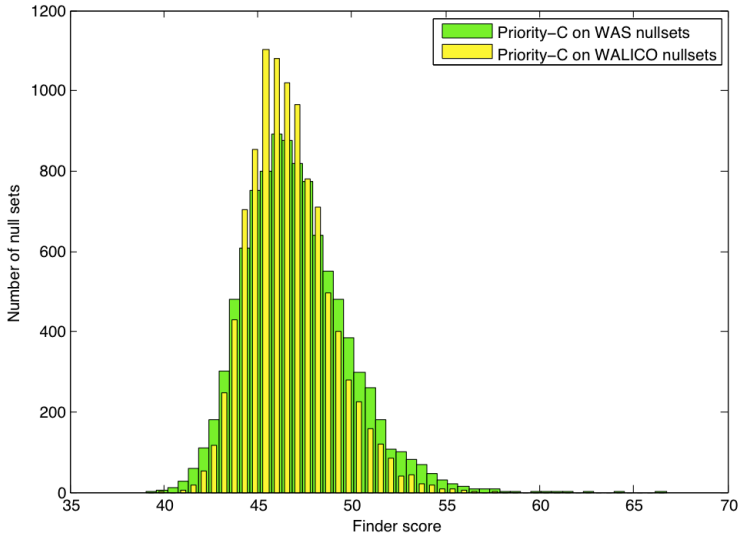


(b) WAS

Figure 9.3: Parametric fit for PhyloCon null distribution. PhyloCon was applied to 10,000 sets of ALICO and WAS resampled alignment-set with GAL4_YPD as template. We were not able to find a parametric fit for PhyloCon under the WAS model. The x-axis is the "total LLR" score returned by PhyloCon.



(a) MEME-C



(b) PRIORITY-C

Figure 9.4: Comparison between WAS and ALICO sampling

Table 9.1: P-values and false-positive rates. The motif finders were run on 156 orthologous sequence-sets. (a) To estimate ALICO-derived p-values, the finders were applied to 50 ALICO-generated null sets for each of the 156 sets. (b) To estimate WAS-derived p-values, the finders were applied to 50 WAS-generated null sets for each of the 156 sets. Note that the values for PhyloCon are obtained from non-parametric evaluation of the p-value under the WAS null model. The false-positive rate (FPR), sensitivity (SEN), number of true-positives (TP), and area under ROC curve (AUC) are reported here.

(a) ALICO-generated null sets

	PhyloCon 3-Gamma			PRIORITY-C (w=8) 3-Gamma			MEME-C (w=8) 3-Gamma			GibbsMarkov (w=8) 3-Gamma		
p-value threshold	FPR	SEN	TP	FPR	SEN	TP	FPR	SEN	TP	FPR	SEN	TP
$p <$ 1E-5	5%	11%	4	3%	41%	27	2%	10%	5	8%	77%	56
$p < 0.01$	8%	32%	12	11%	74%	49	6%	56%	26	40%	92%	67
$p < 0.05$	11%	51%	19	16%	89%	59	14%	79%	36	58%	96%	70
AUC	0.737			0.909			0.886			0.894		
successes (out of 156)	37			66			48			73		

(b) WAS-generated null sets

	PhyloCon Non-parametric			PRIORITY-C (w=8) 3-Gamma			MEME-C (w=8) 3-Gamma			GibbsMarkov (w=8) 3-Gamma		
p-value threshold	FPR	SEN	TP	FPR	SEN	TP	FPR	SEN	TP	FPR	SEN	TP
$p <$ 1E-5	18%	59%	22	1%	29%	19	0%	0%	0	2%	45%	33
$p < 0.01$	18%	59%	22	11%	64%	42	3%	29%	14	10%	73%	53
$p < 0.05$	33%	70%	26	15%	85%	56	7%	56%	27	15%	77%	56
AUC	0.733			0.904			0.889			0.886		
successes (out of 156)	37			66			48			73		

in (Gordân and Hartemink, 2008) (see Section 9.3 for details). Note that these finders do not take actual alignments as input so we had to disregard the alignment itself. PhyloCon had 37 successes, MEME-C had 48 successes, and PRIORITY-C had 66 successes[l]. In addition, the finders were applied to 50 ALICO-generated null sets for each of the 156 sets. To evaluate whether our estimated p-values are well calibrated, we examined the false-positive rates[m] by using "confidence p-values"[n] as the prediction metric (see Table 9.1). While the false positive rates are higher than they ought to be, keep in mind that a false positive here is not necessarily so in the pure statistical sense: some of the "negative" results are due to the finder's detection of a secondary *biologically significant* motif.

While our GibbsMarkov does not fall into the homology-aware finder category it is interesting to see how it performs on the 156 *orthologous* input sets (see Table 9.1). GibbsMarkov is expecting independent input sequences and yet here it is applied to orthologous sequence-sets. Still, our significance analysis yields marginally acceptable control of false-positives (again, recall that a false positive here is not necessarily such from the perspective of significance analysis). Thus, our ALICO sampler offers a method for applying non-homology-aware motif finders to orthologous sets with a somewhat reasonable control of false-positives.

[l]Gordân and Hartemink (2008) reported that PRIORITY-C had 68 successes. The discrepancy is probably due to slightly different compilation of input set and that PRIORITY-C is a stochastic algorithm.

[m]False-positive rate should be approximately equal to the p-value threshold if the p-values are well calibrated.

[n]See Section 6.1 for a review

Table 9.2: Ensemble method. The number of successes were tallied for all possible 512 combinations of GibbsMarkov (w=8) with 9 different finders/widths from PhyloCon, MEME-C, PRIORITY-C, and GibbsMarkov. The number of combinations with corresponding number of successes are shown for selection criteria based on both ALICO-derived and WAS-derived p-values. Note that for each of the 156 sets of each ensemble combination, a single predicted motif was selected based on p-values. Recall that GibbsMarkov (w=8) had 73 successes, which by itself had the highest number of successes among the other 9 different individual finders/widths. See Table 9.3

successes	67	68	69	70	71	72	73	74	75	76	77	78	79
ALICO	0	0	0	0	0	4	80	148	120	72	72	16	0
WAS	1	3	5	3	11	13	61	95	82	97	97	37	8

9.2.5 Using our sampling to combine results from multiple finders

We explore whether our ALICO-derived p-values can be used to improve the motif-finding task by combining multiple motif finders. In particular, we examined the number of successes[o] for all combinations of GibbsMarkov (w=8) with 9 different finders/widths — MEME-C (w=8,13,18), PRIORITY-C (w=8,13,18), GibbsMarkov (w=13,18) and PhyloCon (see Table 9.2). Interestingly out of the 512 combinations, only 4 combinations (less than 1%) are slightly worse off than GibbsMarkov (w=8) alone when using ALICO-derived p-values as selection criteria. Moreover, GibbsMarkov (w=8) has an improvement, i.e. more than 73 successes, 84% (428/512) of the time when combined with other finders/widths via ALICO-derived p-values.

For WAS-derived p-values, there were 7% (36/512) that are worse than GibbsMarkov (w=8) alone, whereas 81% (416/512) had an improvement.

[o]inter-motif distance ≤ 0.25. See Methods for details.

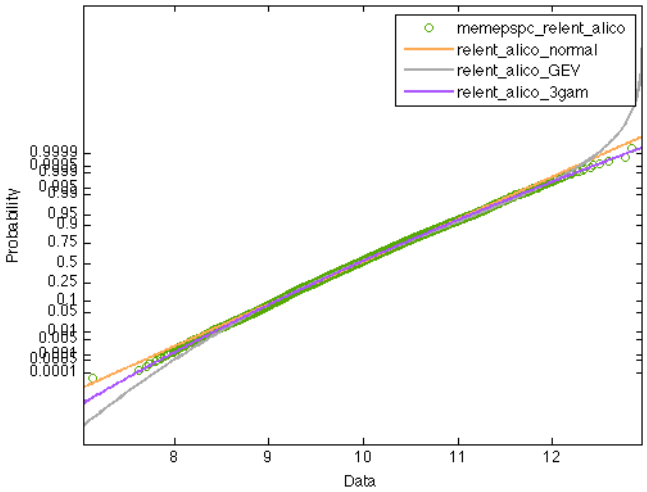
Table 9.3: Performance of individual motif finders

Finder	Width parameter	successes (out of 156)
PRIORITY-$\mathcal{C}$	8	66
PRIORITY-$\mathcal{C}$	13	63
PRIORITY-$\mathcal{C}$	18	52
MEME-$\mathcal{C}$	8	48
MEME-$\mathcal{C}$	13	45
MEME-$\mathcal{C}$	18	27
PhyloCon	N/A	37
GibbsMarkov	8	73
GibbsMarkov	13	73
GibbsMarkov	18	68

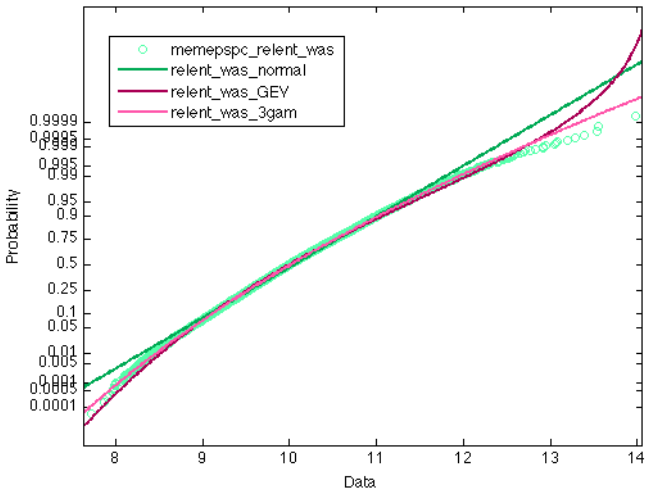
9.3 Methods for ALICO experiments

The Drosophila whole-genome alignments were downloaded from `http://www.biostat.wisc.edu/~cdewey/fly_CAF1/`. The six species used in this chapter were from the melanogaster group, which composed of *D. melanogaster*, *D. simulans*, *D. sechellia*, *D. yakuba*, *D. erecta*, and *D. ananassae*. The template alignment we refer to in the text was generated by sampling segments of these alignments to create a concatenated alignment of length 1 million.

The yeast transcription factor binding data from ChIP-chip, genome-wide location analysis, experiments were obtained from (Harbison et al., 2004). The intergenic orthologous sequences and alignments from 4 species were obtained from (Kellis et al., 2003) http://www.broadinstitute.org/~manoli/yeasts/. The four species used are *S. cerevisiae*, *S. paradoxus*, *S.mikatae*, and *S. bayanus*. The 156 motifs and their literature consensus in our motif-finding benchmark were obtained from (Gordân and Hartemink, 2008). For each of the 156 TF/condition

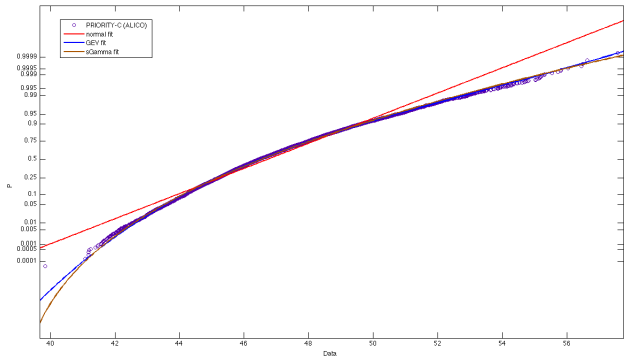


(a) ALICO

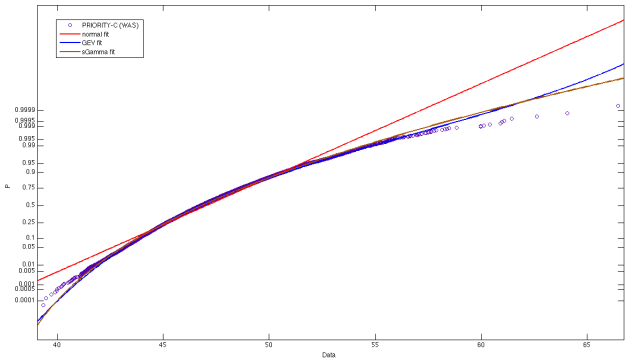


(b) WAS

Figure 9.5: Parametric fit for MEME-C null distribution. MEME-C was applied to 10,000 sets of ALICO and WAS resampled alignment-set with GAL4_YPD as template. The x-axis is the relative-entropy score returned by MEME.



(a) ALICO



(b) WAS

Figure 9.6: Parametric fit for PRIORITY-C null distribution. PRIORITY-C was applied to 10,000 sets of ALICO and WAS resampled alignment-set with GAL4_YPD as template. The x-axis is the score returned by PRIORITY-C.

sets, a motif input set composed of probes that were among the 20 highest binding affinity and binding p-value < 0.001. The inter-motif distance as well as the notion of "success" (distance ≤ 0.25) used in our benchmark is defined exactly as in (Gordân and Hartemink, 2008).

For PhyloCon, all experiments in this chapter used the parameters `"-iq 30 -cq 50 -s 0.5 -o1 -u2 -pc 1 -pt 1"`. We used PhyloCon's "total LLR" as its score in our significance analysis. For GibbsMarkov, the parameters used were `"-cput 300 -L 200 -markov 5 -ds"`. And for Priority-C, all experiments used their default parameters of 50 trials and 10000 iterations. For MEME-C, we used the `hartemink2psp` script included in MEME 4.4.0 to convert conservation-based priors described in (Gordân and Hartemink, 2008) for PRIORITY-C to MEME position-specific priors (PSP). Then we used the MEME parameters "-minsites 10 -dna -revcomp -mod zoops" with 3rd-order Markov model estimated from *S. cerevisiae* intergenic region. We used MEME's "relative entropy" as its score in our significance analysis.

BIBLIOGRAPHY

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–10, Oct. 1990.

S. F. Altschul, J. C. Wootton, E. M. Gertz, R. Agarwala, A. Morgulis, A. A. Schäffer, and Y.-K. Yu. Protein database searches using compositionally adjusted substitution matrices. *The FEBS journal*, 272(20):5101–9, Oct. 2005.

T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings of International Conference on Intelligent Systems for Molecular Biology; ISMB.*, 2(1553-0833):28–36, 1994.

T. L. Bailey and C. Elkan. The value of prior knowledge in discovering motifs with MEME. *Proceedings of International Conference on Intelligent Systems for Molecular Biology ; ISMB.*, 3:21–9, Jan. 1995.

T. L. Bailey, M. Boden, T. Whitington, and P. Machanick. The value of position-specific priors in motif discovery using MEME. *BMC bioinformatics*, 11(1):179, Apr. 2010.

Y. Benjamini and Y. Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289 – 300, 1995.

E. Birney, T. D. Andrews, P. Bevan, M. Caccamo, Y. Chen, L. Clarke, G. Coates, J. Cuff, V. Curwen, T. Cutts, T. Down, E. Eyras, X. M. Fernandez-Suarez, P. Gane, B. Gibbins, J. Gilbert, M. Hammond, H.-R. Hotz, V. Iyer, K. Jekosch, A. Kahari, A. Kasprzyk, D. Keefe, S. Keenan, H. Lehvaslaiho, G. McVicker, C. Melsopp, P. Meidl, E. Mongin, R. Pettett, S. Potter, G. Proctor, M. Rae,

S. Searle, G. Slater, D. Smedley, J. Smith, W. Spooner, A. Stabenau, J. Stalker, R. Storey, A. Ureta-Vidal, K. C. Woodwark, G. Cameron, R. Durbin, A. Cox, T. Hubbard, and M. Clamp. An overview of Ensembl. *Genome Research*, 14(5): 925–928, 2004.

C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268(1):78–94, 1997.

H. Bussemaker, H. Li, and E. Siggia. Regulatory element detection using correlation with expression. *Nature Genetics*, 27(2):167–171, 2001.

J. M. Cherry, C. Adler, C. Ball, S. A. Chervitz, S. S. Dwight, E. T. Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, S. Weng, and D. Botstein. SGD: Saccharomyces Genome Database. *Nucleic Acids Research*, 26(1):73–79, 1998.

G. E. Crooks, G. Hon, J.-M. Chandonia, and S. E. Brenner. WebLogo: a sequence logo generator. *Genome research*, 14(6):1188–90, June 2004.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1 – 38, 1977.

T. A. Down and T. J. P. Hubbard. NestedMICA: sensitive inference of over-represented motifs in nucleic acid sequence. *Nucleic Acids Research*, 33(5):1445–1453, 2005.

R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1999. ISBN 0521629713.

E. Eden, D. Lipson, S. Yogev, and Z. Yakhini. Discovering motifs in ranked lists of DNA sequences. *PLoS Computational Biology*, 3(3):e39, 2007.

W. J. Ewens and G. R. Grant. *Statistical Methods in Bioinformatics: An Introduction (Statistics for Biology and Health)*. Springer, 2004. ISBN 0387400826.

M. C. Frith, U. Hansen, J. L. Spouge, and Z. Weng. Finding functional sequence elements by multiple local alignment. *Nucleic Acids Research*, 32(1):189–200, 2004.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *Journal of Applied Statistics*, 20(5):25–62, 1993.

R. Gordân and A. J. Hartemink. Using DNA duplex stability information for transcription factor binding site discovery. *Pacific Symposium on Biocomputing*, pages 453–64, Jan. 2008.

N. Habib, T. Kaplan, H. Margalit, and N. Friedman. A novel Bayesian DNA motif comparison method for clustering and retrieval. *PLoS Computational Biology*, 4 (2):e1000010, 2008.

C. T. Harbison, D. B. Gordon, T. I. Lee, N. J. Rinaldi, K. D. Macisaac, T. W. Danford, N. M. Hannett, J.-B. Tagne, D. B. Reynolds, J. Yoo, E. G. Jennings, J. Zeitlinger, D. K. Pokholok, M. Kellis, P. A. Rolfe, K. T. Takusagawa, E. S. Lander, D. K. Gifford, E. Fraenkel, and R. A. Young. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104, Sept. 2004.

G. Z. Hertz and G. D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics Oxford England*, 15(7-8):563–577, 1999.

G. Z. Hertz, G. W. Hartzell, and G. D. Stormo. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Computer applications in the biosciences : CABIOS*, 6(2):81–92, Apr. 1990.

J. Hu, B. Li, and D. Kihara. Limitations and potentials of current motif discovery algorithms. *Nucleic Acids Research*, 33(15):4899–4913, 2005.

J. D. Hughes, P. W. Estep, S. Tavazoie, and G. M. Church. Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae. *Journal of Molecular Biology*, 296(5):1205–1214, 2000.

S. T. Jensen, X. S. Liu, Q. Zhou, and J. S. Liu. Computational Discovery of Gene Regulatory Binding Motifs: A Bayesian Perspective. *Statistical Science*, 19(1): 188 – 204, 2004.

N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*, volume 1: Models of *John Wiley Series in Probability and Mathematical Statistics*. Wiley & Sons, 1994. ISBN 0471584959.

S. Karlin and S. F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences of the United States of America*, 87(6):2264–8, Mar. 1990.

U. Keich and P. Ng. A conservative parametric approach to motif significance analysis. *Genome informatics. International Conference on Genome Informatics*, 19: 61–72, Jan. 2007.

U. Keich and P. A. Pevzner. Finding motifs in the twilight zone. *Bioinformatics (Oxford, England)*, 18(10):1374–81, Oct. 2002a.

U. Keich and P. A. Pevzner. Subtle motifs: defining the limits of motif finding algorithms. *Bioinformatics Oxford England*, 18(10):1382–1390, 2002b.

U. Keich, H. Gao, J. S. Garretson, A. Bhaskar, I. Liachko, J. Donato, and B. K. Tye. Computational detection of significant variation in binding affinity across two sets of sequences with application to the analysis of replication origins in yeast. *BMC Bioinformatics*, 9:372, 2008.

M. Kellis, N. Patterson, M. Endrizzi, B. Birren, and E. S. Lander. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, 423(6937):241–254, 2003.

C. Lawrence, S. Altschul, M. Boguski, J. Liu, A. Neuwald, and J. Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262(5131):208–214, Oct. 1993.

J. S. Liu, A. F. Neuwald, and C. E. Lawrence. Bayesian Models for Multiple Local Sequence Alignment and Gibbs Sampling Strategies. *Journal of the American Statistical Association*, 90(432):1156 – 1170, 1995.

X. Liu, D. L. Brutlag, and J. S. Liu. BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pacific Symposium on Biocomputing*, pages 127–38, Jan. 2001.

K. D. MacIsaac, T. Wang, D. B. Gordon, D. K. Gifford, G. D. Stormo, and E. Fraenkel. An improved map of conserved regulatory sites for Saccharomyces cerevisiae. *BMC bioinformatics*, 7(1):113, Jan. 2006.

N. Nagarajan, N. Jones, and U. Keich. Computing the P-value of the information content from an alignment of multiple sequences. *Bioinformatics Oxford England*, 21 Suppl 1(Suppl 1):i311–i318, 2005.

N. Nagarajan, P. Ng, and U. Keich. Refining motif finders with E-value calculations. In *RECOMB on Regulatory Genomics*, 2006.

L. Narlikar, R. Gordân, and A. J. Hartemink. A nucleosome-guided map of transcription factor binding sites in yeast. *PLoS computational biology*, 3(11):e215, Nov. 2007.

A. F. Neuwald, J. S. Liu, and C. E. Lawrence. Gibbs motif sampling: detection of bacterial outer membrane protein repeats. *Protein science a publication of the Protein Society*, 4(8):1618–1632, 1995.

C. A. Nieduszynski, S.-i. Hiraga, P. Ak, C. J. Benham, and A. D. Donaldson. OriDB: a DNA replication origin database. *Nucleic Acids Research*, 35(Database issue):D40–D46, 2007.

A. Prakash and M. Tompa. Statistics of local multiple alignments. *Bioinformatics Oxford England*, 21 Suppl 1:i344–i350, 2005.

A. Price, S. Ramabhadran, and P. A. Pevzner. Finding subtle motifs by branching from sample strings. *Bioinformatics Oxford England*, 19 Suppl 2(90002):ii149–i155, 2003.

R Development Core Team. R: A Language and Environment for Statistical Computing, 2009.

G. K. Sandve and F. Drablos. A survey of motif discovery methods in an integrated framework. *Biology direct*, 1(1):11, Jan. 2006.

T. D. Schneider, G. D. Stormo, L. Gold, and A. Ehrenfeucht. Information content of binding sites on nucleotide sequences. *Journal of molecular biology*, 188(3): 415–31, Apr. 1986.

R. A. Sclafani and T. M. Holzen. Cell cycle regulation of DNA replication. *Annual Review of Genetics*, 41(Figure 1):237–280, 2007.

G. D. Stormo. DNA binding sites: representation and discovery. *Bioinformatics Oxford England*, 16(1):16–23, 2000.

J. A. Swets. Measuring the accuracy of diagnostic systems. *Science*, 240(4857): 1285–1293, 1988.

J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673–80, Nov. 1994.

W. Thompson. Gibbs Recursive Sampler: finding transcription factor binding sites. *Nucleic Acids Research*, 31(13):3580–3585, July 2003.

A. Valouev, D. S. Johnson, A. Sundquist, C. Medina, E. Anton, S. Batzoglou, R. M. Myers, and A. Sidow. Genome-wide analysis of transcription factor binding sites based on ChIP-Seq data. *Nature methods*, 5(9):829–34, Sept. 2008.

T. Wang. Combining phylogenetic data with co-regulated genes to identify regulatory motifs. *Bioinformatics*, 19(18):2369–2380, Dec. 2003.